Machine Learning in Physics project:
Success in Movies
Phase 3: Neural Network

Ali Setareh Kokab , Reyhane Ghanbari

Department of Physics, Sharif University of Technology

June 21, 2021

Contents

1 Summary

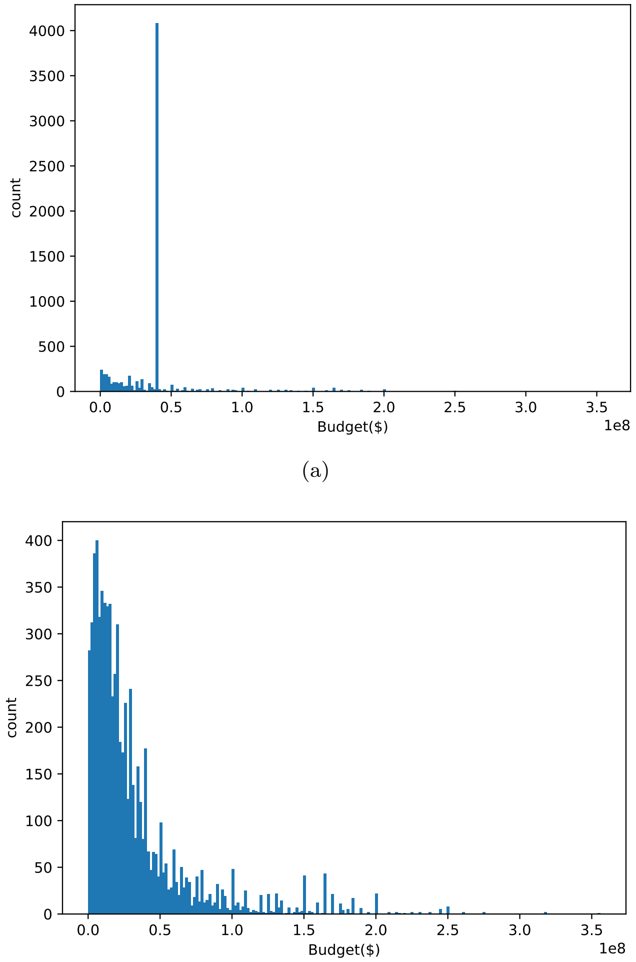
In phase three of our project, we tried neural networks for our regression problem, predicting the success of movies. We also tried to change our problem to the classification problem and apply neural network models to them. We tried two different classification problems. In the first one, we use the worldwide **revenue** of the movies and put the movies in different groups based on this criterion. In the second method, we consider the domestic gross of the movies. We also tried to solve the early prediction of the success of the movies in the box office by considering the first week gross of the movies as one of our features. The structure of this article is as follows. Section 3 represents our results for regression problem using neural networks. Then,4 we describe how we turn our problem from a regression problem to a classification problem. The section 5 represents our effort to tackle the early prediction problem. Finally, in the section 6we conclude our results. In the tables 3 and 4you can see the summary for these models.

2 Preparing the data

For this phase of our project, we tried to focus more on the data preparation part and figure out how to prepare and clean our data more properly. One of the columns with the most missing values (NAN values) in our data set was the budget column which more than 50 percent of its data was missing. This is because most film companies keep their film budget information confidential. To address this problem, we searched on the most famous available movie data sets and added 312 missing budget data to our data set; from these two sources, [1],[2]. Moreover for imputing the missing values, we used KNN imputer with 5 nearest neighbors from sklearn module. The advantage of using this imputer over other conventional imputing methods such as using mean value, is that this imputer does not change the shape of distribution of the variables. in fig 1 you see the difference between imputing with KNN and mean for the budget feature. As it is seen, the knn imputer keeps the structure of the data almost intact. We also used a min-max scaler for scaling our data. As same as the previous phase, we only keep the available features before the movies' release date except for the early prediction section 5 which we use opening gross as one of our features.

3 Regression Problem

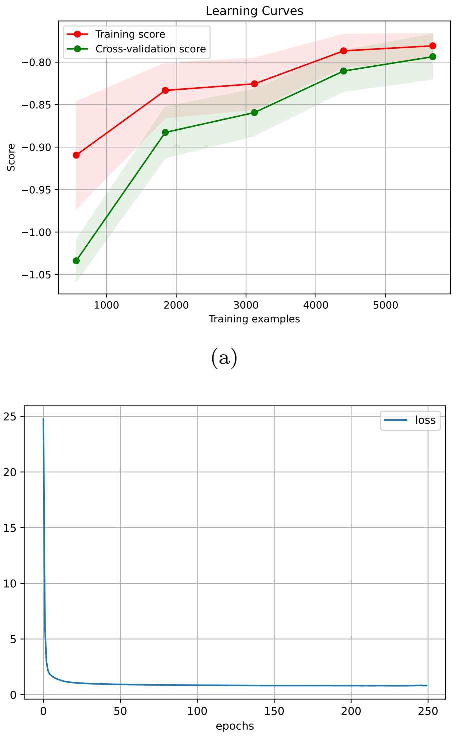
We used sequential model in Keras library to make different dense layers. For choosing the right structure for our neural network, we should consider the size of our data set. As we know, deep neural networks work best on big data sets, and they can cause overfitting on the smaller ones. Hence, due to the small size of our data set and after trying neural networks with different depths, we realized that a network with two hidden layers is the best structure that we can



(a)

(b)

Figure 1: Budget distribution after imputing with a) mean value b) sklearn KNN imputer



(a)

(b)

Figure 2: a) Learning curve for regression problem. It seems that adding more data can do better our model performance. We used negate of mean squared error for the score b) loss versus epochs.

use for our classification problem. Our first layer has 276 inputs which refers to our features. For each layer we have to set node's number and activation function. We choose relu function as an Activation function for first and second layers and as regards to our regression problem the last layer has just 1 node as an output and linear function as an activation function. In fig **??** you see the learning curve for this model.

4 Classification Problem

From the beginning of this project, we have defined our problem as a regression problem. That is it. We wanted to predict an exact number for the box office gross of a movie. In this section we change our point of view by turn our problem into a classification one. We can do this in two ways. One is to classify the movies based on their worldwide revenue. And the other is to consider the domestic gross of the movies regardless of their budget.

4.1 Revenue-based classification

In the Revenue-based classification, we divide the movies into 8 different classes, as described in the table 1.This is the conventional method for dividing the movies in the movie industry.

Table 1: Table of different classes for reveneu based classification

Class	Budget ($)	Worlwide gross ($)
blockbuster	$> 50 \times 10^6$	$> 2.5 \times$ Budget
minor success	$> 50 \times 10^6$	$<= 2.5 \times$ Budget & $>$ Budget
flop	$> 50 \times 10^6$	$<$ Budget
hit	$<= 50 \times 10^6$ & $> 10^6$	$>$ Budget
terrible	$<= 50 \times 10^6$ & $> 10^6$	$<$ Budget & $> 0.1 \times$ Budget
failure	$<= 50 \times 10^6$ & $> 10^6$	$< 0.1 \times$ Budget
great success	$<= 10^6$	$>= 2 \times$ Budget
success	$<= 10^6$	$< 2 \times$ Budget

In fig 3 you see the different classes based on this method. As it is seen in fig 4, most of the movies fall into the failure class. We assign each class a number from 0 to 7. For choosing the right structure for our neural network, we should consider the size of our data set. As we know, deep neural networks work best on big data sets, and they can cause overfitting on the smaller ones. Hence, due to the small size of our data set and after trying neural networks with different depths, we realized that a network with two hidden layers is the best structure that we can use for our classification problem. For the activation functions, we used the tanh function in the first hidden layer followed by relu in the next hidden layer. In the last layer, we used the softmax function for predicting the inputs' class. We also used categorical-cross entropy for the loss function. Finally, we used the Keras tuner to fine-tune the hyperparameters of the network, including each layer's number of nodes and the learning rate. Based on the fine-tuning results, we used 35 nodes for the first layer and 75 nodes for the second layer with a learning rate of 0.01. We also used l2 regularization with the 0.05 regularization constant for each layer.

In fig 5 you see the learning curve and accuracy versus the number of epochs. As it is seen, both of these curves reached to their equilibrium state after 250 epochs.

In fig 6. You see the confusion matrix for this model. As it is seen from these figures, our model works best in detecting hit and failure movies. It also works well in classifying blockbuster movies. The worst performance of our model belongs to the last two classes: great success and success. As it is seen from the table,1 these are very low-budget movies. The reason that our model performs poorly on these two classes is that there are very few of them (compared to other classes) in our data set, and as it is seen in fig 6, in most cases, our model confused these classes with the class failure which are very similar to them.

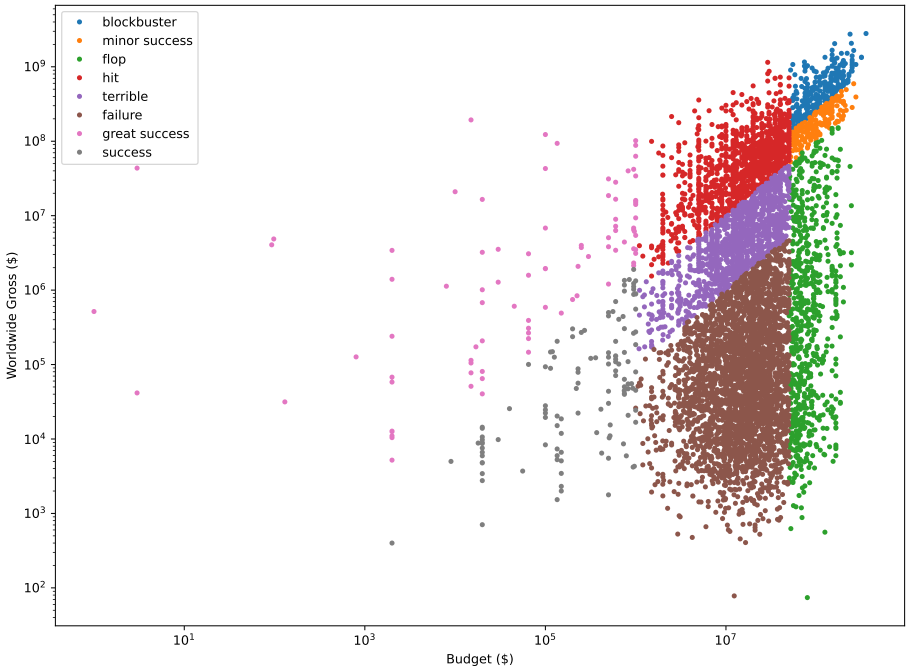


Figure 3: different classes in the classifying the movies based on their revenue.

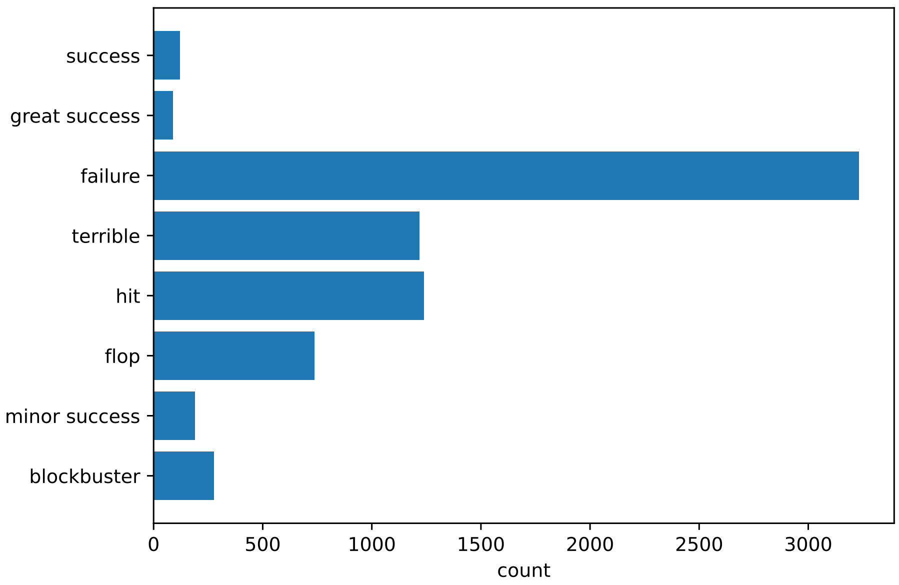


Figure 4: Caption

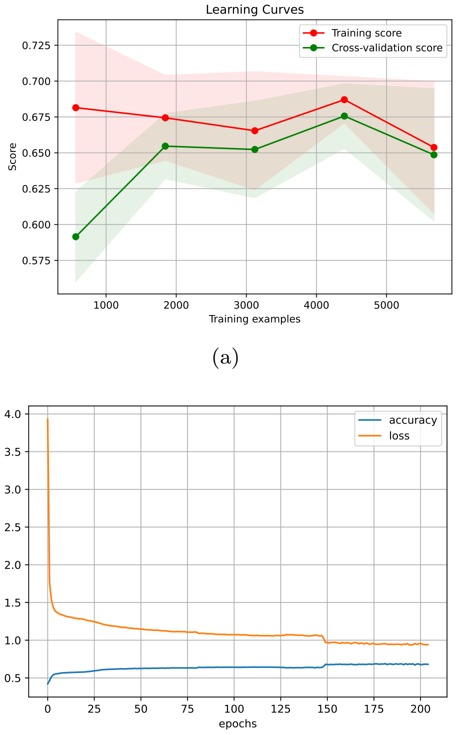
4.2 domestic gross based classification

In the domestic gross-based classification, we divide the movies into 5 different classes, as described in the table 2. In this classification method, we consider the domestic gross of the movies regardless of their budget. As it is seen in figure 7, the domestic gross distribution of the movies has the power distribution. So we divide the movies into 5 equally spaced groups in the log scale.

Table 2: classes based on the domestic gross

Class	domestic gross ($)
very high	$> 9.6 \times 10^9$ & $< 9.3 \times 10^9$
high	$< 9.3 \times 10^9$ & $> 2.1 \times 10^6$
medium	$< 2.1 \times 10^6$ & > 31844
low	< 31844 & > 478
very low	< 478 & > 7

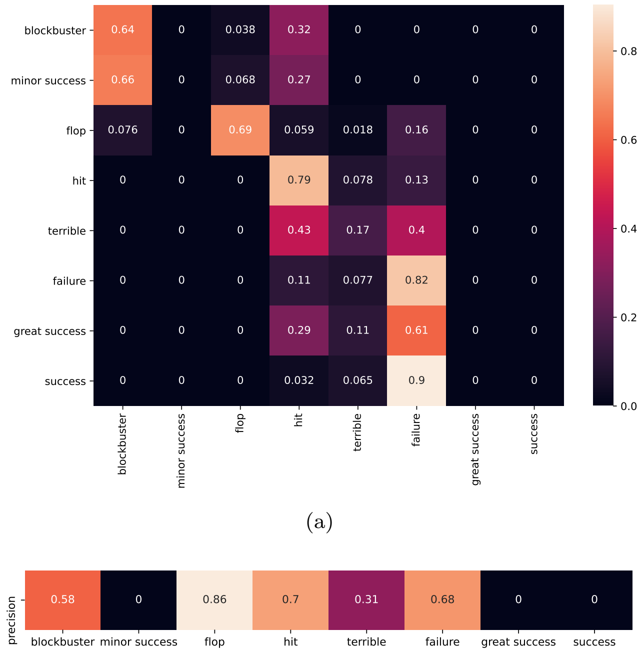
In fig 3 you see the different classes based on this method. As shown in fig 9, most movies fall into the medium class, as is expected. We assign each class a number from 0 to 5. We used the same structure as in the section 4.1 for our neural network, except this time, we used 40 nodes for the first layer and 20 nodes for the second layer. In fig 10 you see the learning curve and accuracy versus the number of epochs. As it is seen, both of these curves



(a)

(b)

Figure 5: a) Learning curve for revenue based classification. As it is seen, both of the curves reached to their equlibrium values. This means that adding more data to our data set would not result in significant increase in performance of the model. b) cross-validation curve for this neural network. Both loss and accuracy reached to their final values after 250 epochs.



(a)

(b)

Figure 6: a) confusion matrix for revenue based classification b) precision for each class

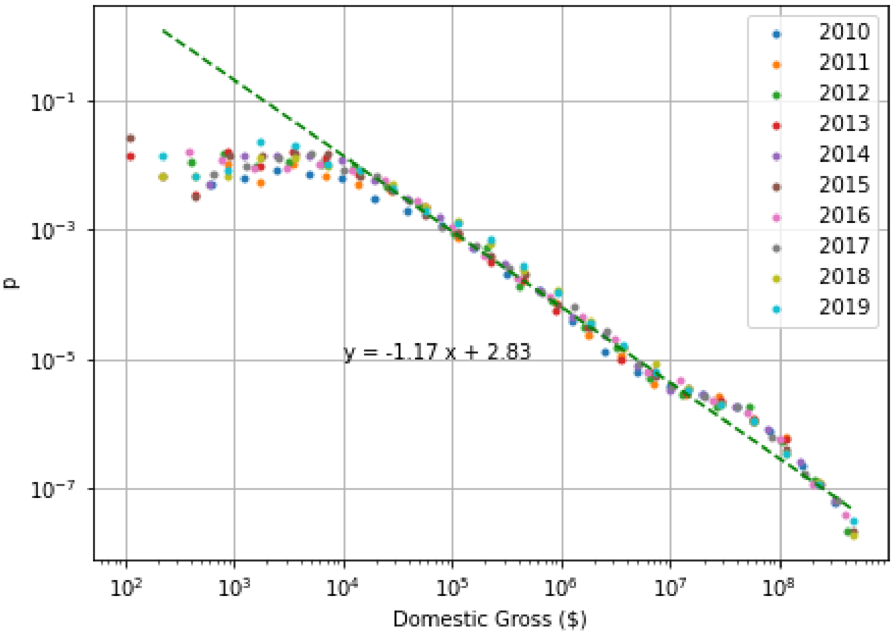


Figure 7: domestic gross distribution of the movies for different years. As it is seen, this distribution has a power law form. The power is equal to -1.1.

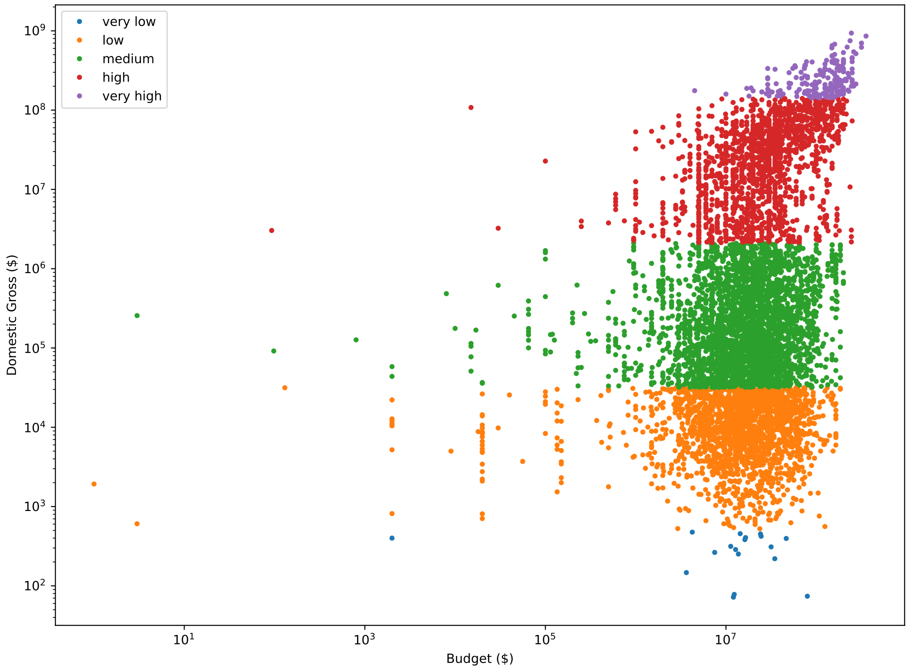


Figure 8: different classes in the classifying the movies base on their domestic gross.

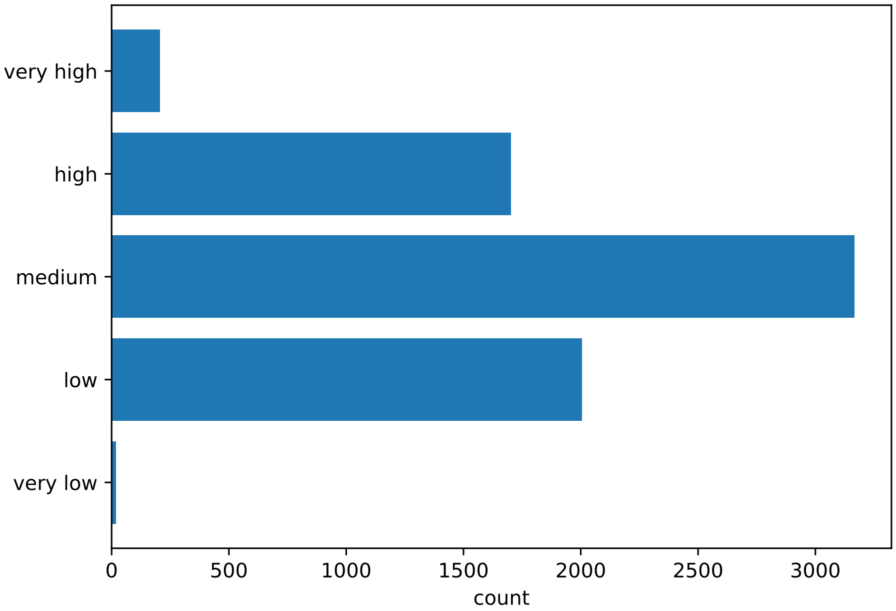
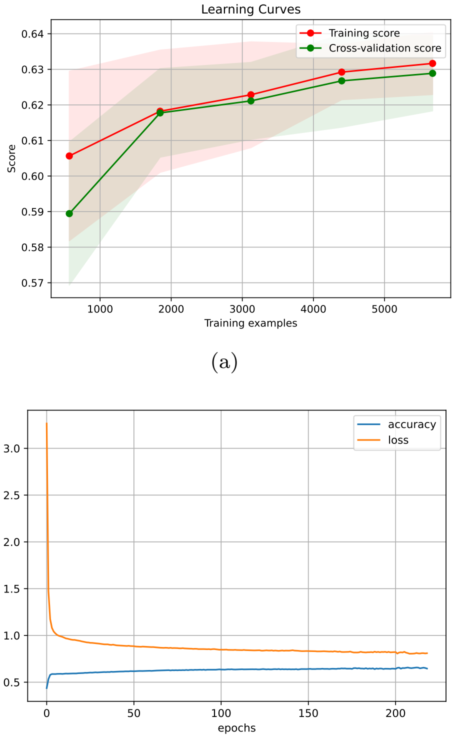


Figure 9: distribution of different classes for domestic based classification.

reached to their equilibrium state after 250 epochs. In fig 10 you see the learning curve and accuracy versus the number of epochs. As it is seen, both of these curves reached to their equilibrium state after 250 epochs. In fig 11. You see the confusion matrix for this model. As seen from these figures, our model works best to detect movies with medium and high box office gross. The worst performance of our model belongs to the last movies with very high gross and very low gross. As shown in fig 11, our model mostly confusing movies with high domestic gross with very high domestic gross. In fig,12 you see the distribution predicted by our model compared to real distribution.

5 Early prediction

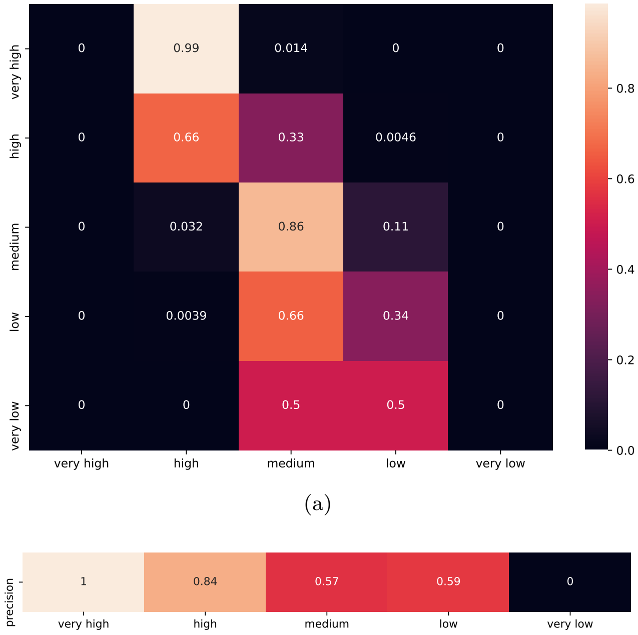
So far, we have just considered the features that are available before the movies' release data. In this section, we want to examine the case, which is called early prediction. For this problem, we won't predict the success of the movies by having the before-release data of a movie and having some data of the first week of a movie. So in this section, we also consider the opening gross as one of our features. The opening gross is the gross of a movie in its first week of release. We also consider domestic gross as our target variable. We solve this problem both for regression and classification.



(a)

(b)

Figure 10: a) Learning curve for domestic based classification. As it is seen, it seems that adding more data could help our model to perform better. b) cross-validation curve for this neural network. Both loss and accuracy reached to their final values after 250 epochs.



(a)

(b)

Figure 11: a) confusion matrix for gross based classification b) precision for each class

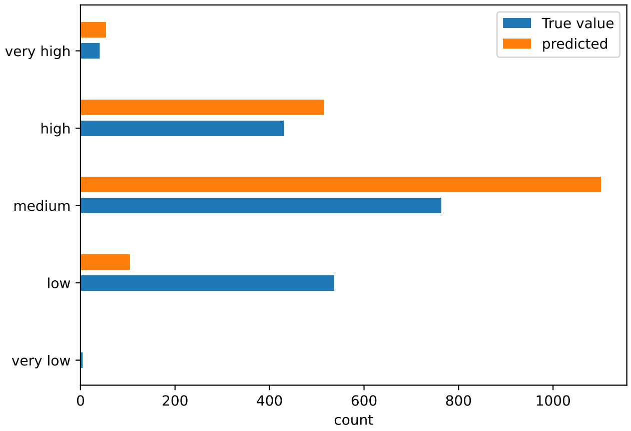
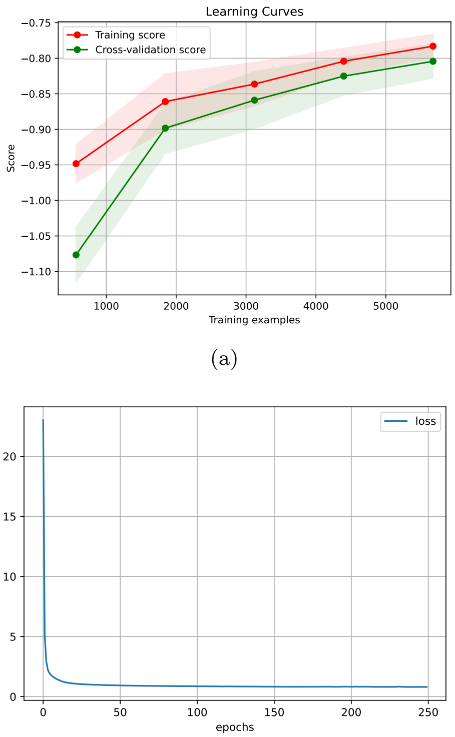


Figure 12: predicted distribution by neural network compared to real distribution



(a)

(b)

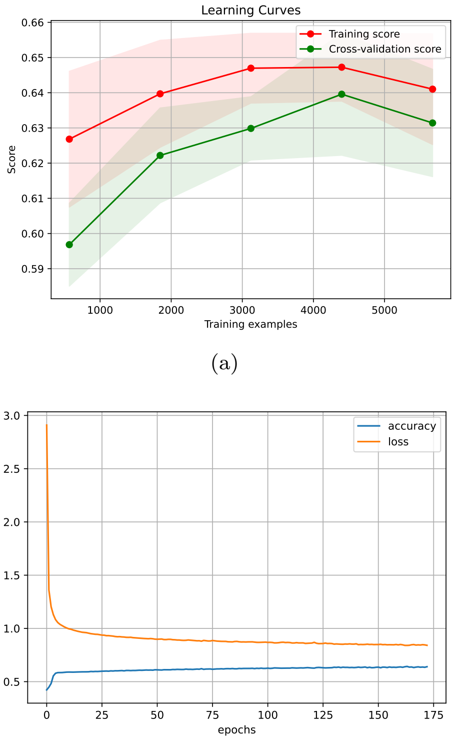
Figure 13: a) Learning curve for domestic gross early prediction regression problem. We used negate of mean squared error as our score.b) loss versus epochs

5.1 Regression

For the regression problem, we consider domestic gross as the target variable as before. After fine-tuning, we used a two-layer network with 10 nodes in the first layer and 25 nodes in the second layer. In both of these layers, we used relu activation function. For the loss function, we used mean squared error (mse). In fig 13 you see the learning curve and loss versus epochs.

5.2 Classification

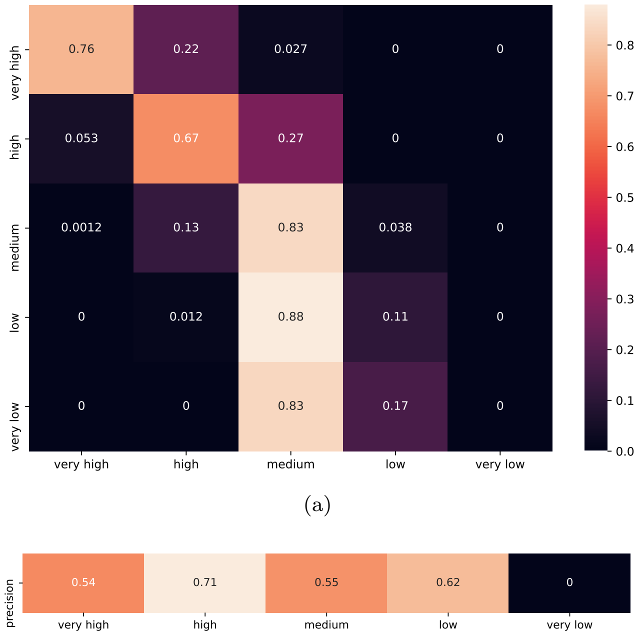
We classified the movies as it is described in 2. Our network structure is the same as 4 with 10 nodes in the first layer and 85 nodes in the second layer. In fig. Y,14 you see the cross-validation curves and learning curves for this model. In fig. Y,15 you can see the confusion matrix for this model. As it is seen, our performance is slightly better here in comparison with models in 4, especially for classifying the high gross movies, which now we have the precision 0f 0.71.



(a)

(b)

Figure 14: a) Learning curve for domestic gross early prediction classification.



(a)

(b)

Figure 15: a) confusion matrix for early gross based classification b) precision for each class

6 conclusion

In this phase, we try to predict the movies' success using neural networks. We tackle both regression and classification problems. For the classification problem, we used two different schemes based on the worldwide revenue and the other based on the domestic gross. We used multiple tools from sklearn and TensorFlow libraries such as Keras tuner and KerasClassifier to tune and evaluate our models. We also examine the early prediction problem by using the first week gross of the movies as one of our features. In the table 3 and 4 you see our results.

Classification	Average Accuracy	Average Precision	Learning time (s)	Prediction time (s)
Revenue based	0.69	0.6	26.4	0.021
Domestic gross based	0.63	0.62	22.3	0.034
Early domestic gross based	0.64	0.61	25.1	0.041

Table 3: Summary of results for classification problem

Regression	mse on train	r2 score train	mse on test	r2 score on test	Learning time (s)	Prediction time(s)
Regular	0.70	0.63	0.73	0.62	32	0.021
Early	0.64	0.66	0.7	0.63	24.1	0.026

Table 4: Summary of results for regression problem

Appendices

A Metrics

we use mean squared error and R^2 (1) score to evaluate our models. These are widely used metrics for regression problems. R^2 is a **relative** measure of how well the model fits dependent variables and measures how the model can explain much variability in the dependent variable. Its value is between 0 and 1, and a bigger value indicates better performance.In eq 1, $\hat{y}$ and $\overline{y}$ represents the predicted and real values, respectively.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \overline{y})} \tag{1}$$

We also used mean square error (2) to evaluate the performance of our models. Despite the R^2 score, mean squared error is an absolute measure of the goodness of our fit. So it is a useful metric to compare different regression with each other and check the performance of neural network model compare with traditional models.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{2}$$

For classification problems, we used accuracy and precision to evaluate our models. Because our problem has multi classes we calculate the precision for each class and getting the weighted average of them.

References

[1] The movies dataset — kaggle. https://www.kaggle.com/rounakbanik/the-movies-dataset. (Accessed on 06/20/2021).

[2] Tmdb movies dataset — kaggle. https://www.kaggle.com/juzershakir/tmdb-movies-dataset. (Accessed on 06/20/2021).