**MOVIE CLASSIFICATION DATSET**

Allie Baker, [abaker8@bellarmine.edu](mailto:abaker8@bellarmine.edu)

Nik Mrdak, [nmrdak@bellarmine.edu](mailto:nmrdak@bellarmine.edu)

**ABSTRACT**

For our final project, we use a Movie Classification data set from Kaggle. We used this data set because of the interests both have in movies. We wanted to see if our model could reasonably predict how much money a movie will collect and whether a movie will be awarded an Oscar or not. Within this document, you will find visualizations of our data and the variables used. You can also find the linear and logistic regression descriptions that allowed us to find the most accurate results for our predictions.

1. **INTRODUCTION**

Provide a one or two paragraph introduction to your project in which you describe the data set you are working with and the classification target (what are you trying to predict?) you will be pursuing.

Our dataset we chose is a movie classification dataset. The dataset offers a wide range of comprehensive information essential for movie enthusiasts, analysts, and researchers. It comprises an array of critical details including production expense, budget, genre, and whether or not it won an Oscar. We wanted to see if our model could reasonably predict how much money a movie will collect and whether a movie will be awarded an Oscar or not.

1. **BACKGROUND**

In this section, provide some background for the problem for which the data were collected. For example, if you were using the mushroom data set, you write up some background on what a mushroom is, why the data were originally collected, what question(s) the authors were trying to answer, etc.

Movies are a significant part of our culture and entertainment, encompassing a wide range of genres such as action, drama, comedy and thriller. The classification of movies into genres is crucial for various applications such as content filtering and marketing suggestions. The primary purpose of collecting movie classification data is to develop models that can accurately predict whether a movie would earn an Oscar or not. Factors such as movie views, ratings, and genre are all important factors. A question the authors may have aimed to answer is: What features are most influential in determining a movie’s genre and whether it would win an Oscar? Overall, this dataset can help to automatically categorize movies to help users find content that matches their preferences and can help the analysts understand genre trends to tailor marketing campaigns effectively.

1. **EXPLORATORY ANALYSIS**

This section will be similar to your exploratory analysis project. First, provide a summary of the data set similar to your first exploratory analysis: *e.g. this data set contains 398 samples with 7 columns with various data types*. In this summary, provide the data types of your columns (in a table) and then rather than providing tabular statistics and plots for each variable, provide only statistics and plots that seem unusual. For example, if one or two variables have significant missing values or the distribution of the variable is skewed or looks unusual note that. Provide the unusual statistics or plots in this section. Provide any other appropriate plots (e.g. correlation matrix, heatmaps, bar charts, etc.) that you deem necessary.

This data set contains 506 samples with 19 columns with various data types. The Time\_taken variable has 12 missing values; to solve this issue, we filled the missing values with the mean. Most of our data is evenly distributed. One thing that is interesting to note, is in our correlation matrix, there was not a high correlation between movie figures and obtaining an Oscar. In addition, Variables 6-9 displayed multicollinearity, which we predicted would affect the accuracy of our models.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| V1 Marketing expense | Float64 |
| V2 Production expense | Float64 |
| V3 Multiplex coverage | Float64 |
| V4 Budget | Float64 |
| V5 Movie\_length | Float64 |
| V6 Lead\_Actor\_Rating | Float64 |
| V7 Lead\_Actress\_Rating | Float64 |
| V8 Director\_Rating | Float64 |
| V9 Producer\_Rating | Float64 |
| V10 Critic\_Rating | Float64 |
| V11 Trailer\_Views | Int64 |
| V12 3D\_Available | Object |
| V13 Time\_Taken | Float64 |
| V14 Twitter\_Hashtags | Float64 |
| V15 Genre | Object |
| V16 Avg\_age\_actors | Int64 |
| V17 Num\_multiplex | Int64 |
| V18 Collection | Int64 |
| V19 Start\_Tech\_Oscar | Int64 |

1. **METHODS**

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model(s).

* 1. *Data Preparation*

Describe how you prepared the data for your model. For example, you might need to normalize the data, so variables with wider ranges of values don’t overshadow variables with smaller ranges. If you decide to drop variables from the model or create variables from existing columns, explain the process and the reasoning behind those decisions.

First, we checked the first couple rows of the data to understand our dataset. We found the count, mean, standard deviation, minimum, maximum, Q1, Q2, and Q3. There were a couple of missing values in this dataset so we filled them with the mean. We chose to use the mean based off the feature distribution. We coded the Genre column and 3D available column using get\_dummies to be True or False. This was because the data type for these columns was an object, and we wanted all of our data to be float64 or int64.

* 1. *Experimental Design*

You will run your model several times with different parameters to see what different results you get. In a table, describe your experimental parameters. Three or four experiments are sufficient. This is where you will describe how you divided your data into train, validate and test data sets. For example:

**Table X: Experiment Parameters MLR**

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | In our MLR, all variables excluding V6-V9, with 80/20 split for train, and test |
| 2 | Same excluded features with 75/25 split, changed random state to 5 |
| 3 | All variables including V6-V9 with 80/20 split gave best results with r.s. 5 |

**Table XI: Experiment Parameters Logistic**

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | In our Logistic model, all variables with 80/20 split for train, and test |
| 2 | Same excluded features with 75/25 split, changed random state to 61 |
| 3 | All variables excluding V6-V9 with 80/20 split gave best results with r.s. 61 |

* 1. *Tools Used*

Describe all of the software tools you used to perform your data preparation and model implementation. For example:

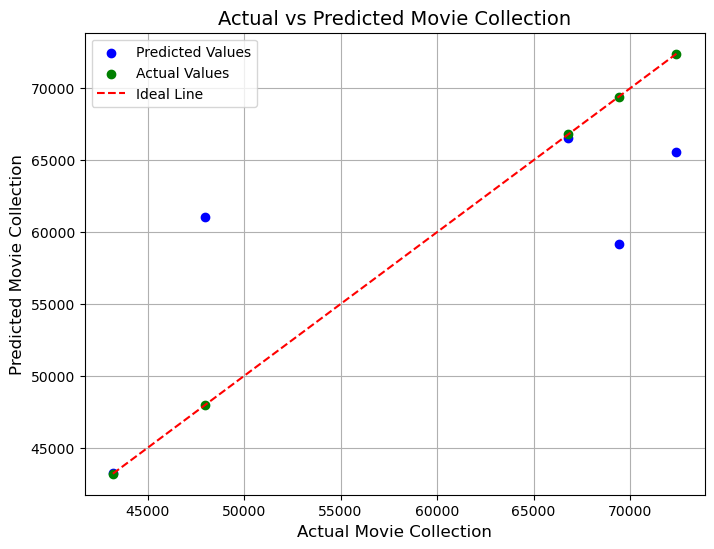
The following tools were used for this analysis: Python v3.5.2 running the Anaconda 4.3.22 environment was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, and SKLearn 0.18.1. Pandas is essential for data manipulation and analysis. It provides data structures that are perfect for handling and analyzing structured data. Numpy is needed for numerical computations. It supports large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. Mathplotlib is a plotting library used for creating static, interactive, and animated visualizations in Python. Seaborn provides a high-level interface for drawing attractive and informative statistical graphs, creating complex visualizations. SKLearn includes simple and efficient tools for fata mining and data analysis, making it ideal for building and evaluating machine learning models.These tools and libraries were chosen because they are widely used in the data science community and are functional for data analysis and machine learning tasks.

1. **RESULTS**
   1. *Classification Measures/ Accuracy measure*

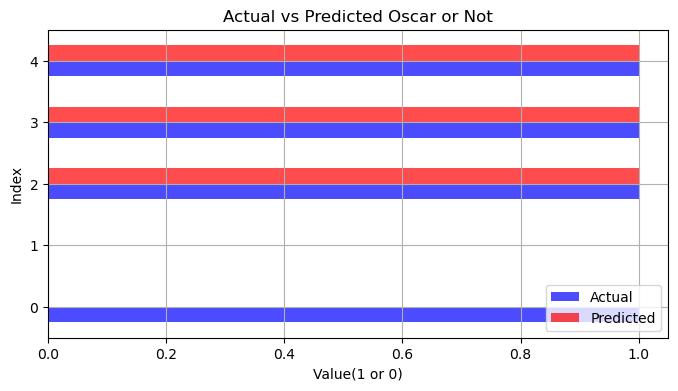
Provide the classification/ regression measures for each experiment. For example, you could provide a contingency table for each model to measure how well it classifies data. You could also do an ROC curve (using SciKit Learn). You need to demonstrate how you are measuring the success/failure of the models. For regression, you can calculate the RMSE and scores and compare between various models.

The method we used to measure the accuracy of our models was to preserve five rows from the data from the train / test split, so that we could impute those rows into the model and compare the predicted values to the actual values given. Below are two visualizations comparing the predicted and actual values. Overall, the model makes moderately accurate predictions but struggles with outliers.

**Figure 1: MLR Predicted v. Actual**



**Figure 2: Logistic Classification Predicted v. Actual**



* 1. *Discussion of Results*

Discuss which of your models provided the best classification (or some other outcome if not classification). Explain why you think your best model was the best and why your worst model was the worst.

We believe that the logistic model performed the best classification, as it accurately predicted 4 of 5 target variables from the validation data. The MLR model adequately predicted collection for movies, but struggled more with outliers than the logistic model did. The first presumption was that this is caused by variables 6-9 being excluded from the logistic model but not from the linear model; however, the linear model performed better with those variables included than without.

* 1. *Problems Encountered*

No project goes perfectly smooth. Discuss any problems you had with obtaining the data, preparing the data, implementing the model, or evaluating the model. **It would be highly unusual to indicate that you had no problems.**

One of our issues with both linear and logistic models was implementing and improving them. Dr. Sarkar gave us the direction we needed to correct our models to get them running correctly, as well as improving the accuracy of our models with a method to compare the predictions our models make with actual values given in the data.

* 1. *Limitations of Implementation*

Discuss the limitations of your model(s). Are there reasons your models might not be the best way to predict the target data? What other models might work better?

One reason the model might not be the best at predicting the target variables is the limitations of the data itself. More variables and rows within the data may provide a more accurate regression model. Furthermore, given the data contained more Oscar winning movies than non-Oscar winning movies, the unequalness of the two classes may have been cause for lower statistical values in the classification report.

* 1. *Improvements/Future Work*

What would you like to do to improve your model in future work? Some items you might consider discussing are performing more experiments, using different models, adding or removing variables, finding a different data set, etc.

To improve the model, we would be interested in formulating a larger dataset for these models to train from, as well as incorporating a more advanced model. These two features increase the likelihood of more accurate predictions made and improve the overall quality of the model.

1. **CONCLUSION**

Finish up with a paragraph or two of summarizing your problem, the results and your conclusions (good model, bad model, needs more work, etc.).

The goal of this project was to be able to reasonably predict how much money a movie will collect and whether a movie will be awarded an Oscar, given the parameters of the movies classification dataset. The EDA we conducted prepared our data for the machine learning models by encoding missing values and relabeling object datatypes as Boolean. The biggest issue with the linear regression model we faced was a perfect 1.0 R2 score. To double check the validity of our model’s predictions, we preserved five rows from the training set to use for comparing the actual values of collection and the one the model predicts from the values of the reserved rows. After applying this method with minor tweaks to the model, the r2 score changed to a more realistic 0.82. The biggest issue that the logistic model presented was the high number of type I and type II errors within the confusion matrix. To improve the accuracy and precision we changed the random state score as well as removing for variables that had high multicollinearity with one another. Overall, the final state of the two models generates adequate predictions on the movies dataset, but improvement could be made with scaling the amount of data to train the model with.

**REFERENCES**

None used.