Using User Tags to Classify Movie Genres

[1]Paul Verbovshchuk, [2]Michael Haight, [3]Gadfrey Balacy

Computer Science, California State University, Sacramento
6000 J St, Sacramento, CA 95819

[1]verbovshchuk@csus.edu

[2]michaelhaight@csus.edu

[3]balacy@csus.edu

Abstract— **Human labor is typically required in the process of movie genre determination. This can be alleviated, as the mass of information available on the internet can be used to create intelligent predictions without the need for a specific genre classifier role. We propose a way to classify movie genres by analysing movie feedback provided by users. User comments were processed and inserted as inputs into Logistic Regression, Decision Tree, Neural Network, and Support Vector Machine learning algorithms to determine a link between user reviews and genre classification. We trained each algorithm on the Action, Adventure, Comedy, and Drama genres. Our results indicated that the Neural Network algorithm outperformed the others when the genre included a large amount of tags. In all other scenarios, Logistic Regression featured the highest accuracy.**

Keywords— **Movie genre classification, Logistic Regression, Decision Tree, Neural Network, Support Vector Machine**

I. Introduction

A. Our Motivating Problem

Modern movie classification processes are resource draining and lack efficiency. Hiring professional movie critics can be a burden on the financial health of a production company, and manually predicting movie genres by sifting through reviews is tedious and time consuming. To the best of our knowledge, there is not a current genre classifier system in place. In this study, we attempt to classify movies into different genres by using movie reviews collected from the internet.

B. Overview of Our Approach

Our paper presents an approach to categorizing movies into various genres. We filtered a collection of 750 thousand tags written by 270 thousand users into a smaller samples of data to reduce the training time for each set, but kept it large enough where accuracy is not impacted significantly. After processing, we ran Logistic Regression, Decision Tree, Neural Network, and the Support Vector Machine learning algorithms with the tags as input for each genre and compared the effectiveness of each algorithm.

C. Our Contributions

We summarize our contributions as follows:

- We discuss methods for filtering tags from movie review websites to increase prediction accuracy.
- We provide a way to apply training and testing algorithms to data relevant to the movie domain.
- We provide a model for measuring the performance of several machine learning algorithms based on samples of tags.

D. Paper Organization

The rest of the paper is organized as follows. The research problem is formally defined in Section II. In Section III, we introduce the system architecture and discuss the algorithms that were used to train and test the data. Section IV describes the experimental evaluation, including the process for data splitting, the methods for algorithm design and metric analysis, and the results of the trials. Section V reviews related works. Section VI briefly summarizes the results and conclusions. Section VII provides an overview of how the work was divided within our group. Section VIII concludes with an analysis of what we have gained through doing this study.

II. Problem Formulation

Our problem addresses the categorization of movies into genres. We aim to predict the associated genres of a movie by collecting user reviews and inferring based on their language. The input we used are the movie reviews left by users obtained from a movie review database. The outputs are either a true or false variable for each of the four genres that we used: Action, Adventure, Drama, and Comedy. The data set had 19 genres initially, but we cut the outputs for brevity. If necessary, this can be expanded to include more genres.

III. System Design

The system contained the following stages: pre-processing, training, and testing. Filtered tags that were obtained from pre-processing were fed into the Logistic Regression, Decision Tree, Neural Network, and the Support Vector Machine algorithms and then evaluated based on defined metrics. The results from training were compared with the results from testing against ground truths for each of the tested genres.

A. Logistic Regression

Logistic Regression creates different class groups with the predicted probability. It can be used to put bounds on the output, as it will output a value between 0 and 1. In cases where it is used as a binary classifier, the probability threshold

can be set to 0.5. If the predicted value falls below 0.5, the output will be 0 and if it is greater than or equal to 0.5, the output will be 1. It serves as a good model when output values between 0 and 1 are appropriate.

TABLE I
SAMPLE DATA FOR LOGISTIC REGRESSION ALGORITHM [1]

Studied	Slept	Passed
4.85	9.63	1
8.62	3.23	0
5.43	8.23	1
9.21	6.34	0

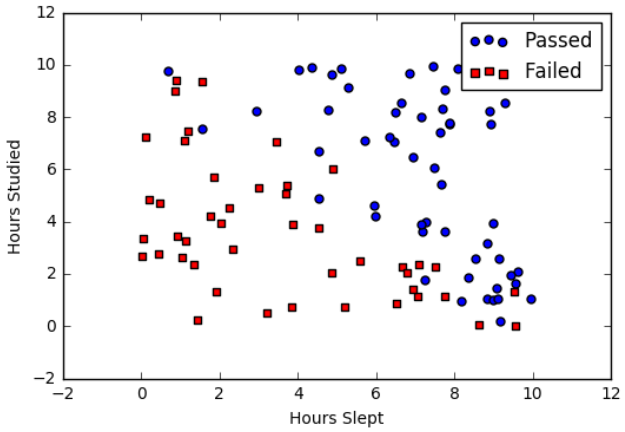

Fig. 1 Scatter plot of Table I [1]

Table I shows the sample test result values for students with information of the number of hours of sleep they got and the amount of hours they spent studying. Fig. 1 shows a scatter plot of these values. After running the data through logistic regression, a decision boundary line would be drawn diagonally where the passed and failed points meet.

B. *Decision Tree*

The goal of the decision tree algorithm is to create a training model from learned decision rules. The rules are created from training data, and are then used to predict the value of a class. The decisions use the tree structure where each internal node represents a test on an attribute, each branch denotes the outcome of the test, and each leaf node represents class labels.

The basic pseudocode is as follows:

1. The best attribute is placed at the root of the tree
2. The training set is split into separate subsets. Each subset contains data with the same value for the attribute.
3. Steps 1 and 2 are repeated until the leaf node is found for each branch of the tree [2].

TABLE II
TESTING DATA FOR TAX FRAUD DETECTION [3]

Refund	Marital Status	Taxable Income	Cheat
No	Married	80k	?

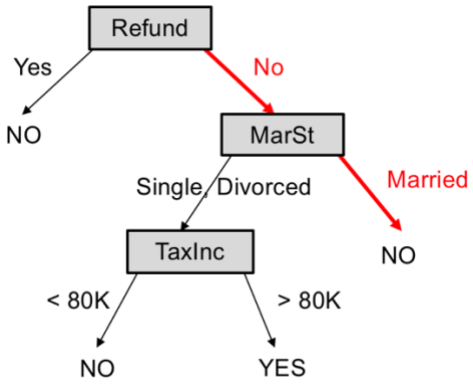

Fig. 2 Decision Tree for predicting value of "Cheat" value in Table II [3]

Fig. 2 demonstrates the path that the Decision Tree model will take when predicting the value for "Cheat" in Table II. The algorithm will ultimately predict that the case demonstrated in Table II is not an instance of tax fraud, and will guess "No."

C. *Neural Network*

A Neural Networks is a collection of neurons, each of which has an associated weight. The simplest form of a Neural Network is equivalent to linear regression. If a hidden layer is present, then the outputs of the first layer are combined using a weighted sum and fed in as inputs to the next layer. A nonlinear function is applied to the neurons and an output is generated. If the output does not match the ground truths during training, the weights are adjusted and the process is done again. Once the accuracy is at a satisfactory level, the information gained from training can be used to predict test data.

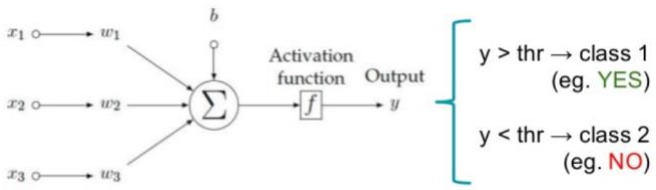

Fig. 3 Simple Neural Network with one neuron [4]

Fig. 3 shows a simple network consisting of just one neuron. There are three inputs, and each of them is attached to a weight before it is summed and passed into an activation function, which does the final processing before an output is generated. Depending on thresholds set, the output is then classified as either the first or second class.

D. *Support Vector Machine*

Support Vector Machine is an algorithm design to find the broadest linear boundary between two different classes. It is optimized to work on data that is linearly separable, as it creates multiple plains and uses the plain with the most significant margin between the different classes.

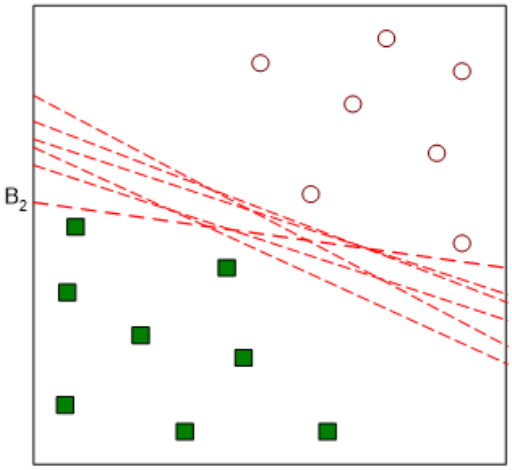

Fig. 4 Example of SVM with multiple plains dividing classes [5]

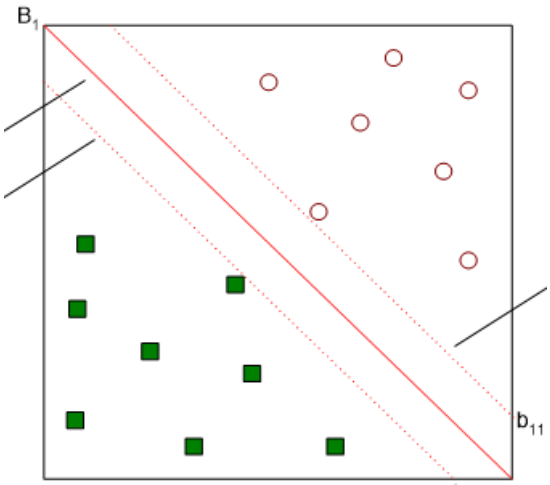

Fig. 5 SVM with plane dividing classes at the widest margin [5]

Fig. 4 shows the SVM algorithm as it is performing it's optimization. It has multiple plains diving the data points, and is in the process of figuring out which plain best divides the data points. Fig. 5 shows the final result of the divisibility line optimization, with the line that has the widest margin between both classes having been chosen as the final plain.

IV. Experimental Evaluation

A. Methodology

We used the full 26 million ratings MovieLens database provided online by GroupLens [6]. We combined all of the tags for each corresponding movie together to help filter out movies that did not contain enough information to train and test the genres. Movies with less than 20 tags were removed, and users who did not contribute any reviews were removed as well. Movies whose genre was unknown were also removed. The tags were vectorized as features, and the parameters were tuned to ignore tags that had less than 20 occurrences. After pre-processing, 5057 movies and 2178 tags remained.

The outputs genres were separated and cut from 19 separate genres to 4, for brevity. The final genres we used were Action, Adventure, Drama, and Comedy. Those genres were selected because they were the most popular genres for movies to be classified as, but the methods we used could be expanded to all of the other genres as well. The genres we selected were encoded as either 0 or 1 for each movie, depending on whether each movie was classified as such. For each category, the training and testing sets were split at 80% training and 20% testing, with 4045 movies allocated for training and 1012 for testing. For each genre, we processed the data through the SVM, Logistic Regression, Decision Tree, and Neural Network algorithm.

Standard algorithms were run for SVM, Logistic Regression, and Decision Tree. After trial runs with the standard algorithms, Gridsearch was utilized to find the optimum parameters. The parameters for Decision Tree were found to optimally be at entropy for the criterion. The C values were changed for the SVM algorithm on each genre. However, the Logistic Regression algorithm was not modified by the Gridsearch. The Neural Network parameters were manually tuned to use the "tanh" activation function, "sgd" as the solver, a momentum of 0.9, 30 x 30 x 30 hidden layer sizes, and 1000 max iterations.

B. Results

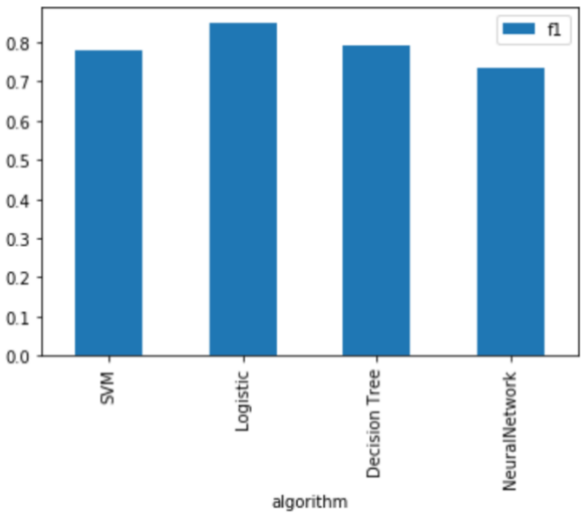

Fig. 6 F1 scores of Comedy genre, before Gridsearch

Fig. 6 shows the F1 scores of the SVM, Logistic Regression, Decision Tree, and Neural Network algorithms on the Comedy genre.

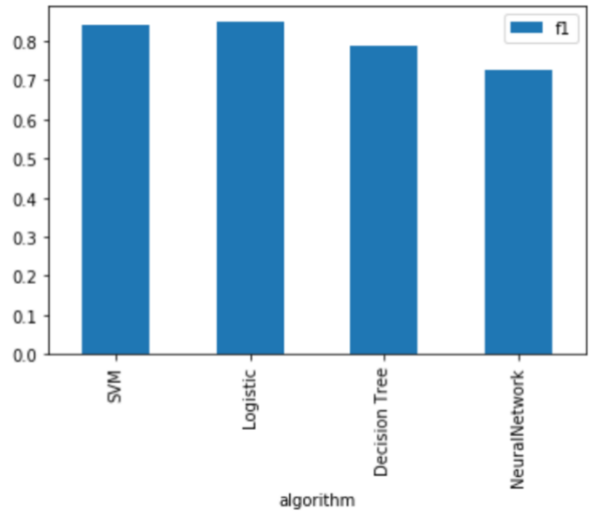

Fig. 7 F1 scores of Comedy genre, after Gridsearch

In order to enhance accuracy of the SVM, Logistic Regression, and Decision Tree results, we implemented Gridsearch into each of those algorithms. Fig. 7 shows the F1 scores of each algorithm on the Comedy genre after applying Gridsearch. Gridsearch's parameter tuning capabilities visibly enhanced the accuracy of the SVM algorithm, and also slightly improved the Logistic Regression and Decision Tree algorithms. Even though the above charts only show the results for Comedy, this trend was visible in all of the genres tested.

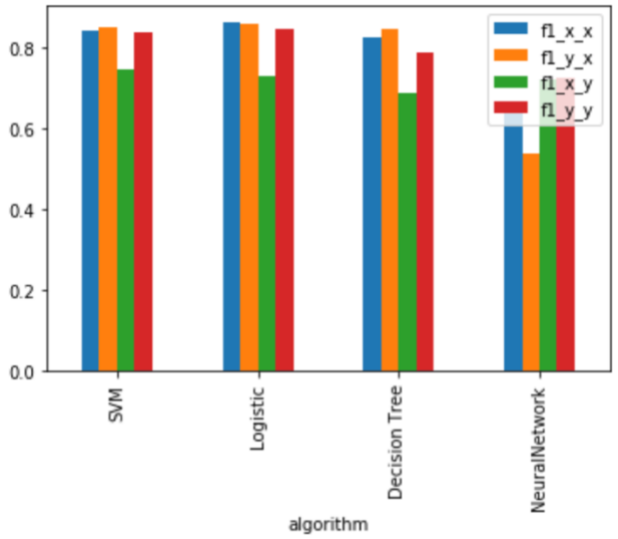

Fig. 8 F1 scores of each algorithm on each genre after Gridsearch

	algorithm	Action	genre_x_x	Adventure	genre_y_x	Drama	genre_x_y	Comedy	genre_y_y
0	SVM	0.843202	Action	0.851463	Adventure	0.745066	Drama	0.839564	Comedy
1	Logistic	0.863068	Action	0.860196	Adventure	0.732024	Drama	0.848482	Comedy
2	Decision Tree	0.824097	Action	0.846515	Adventure	0.687743	Drama	0.787319	Comedy
3	NeuralNetwork	0.636569	Action	0.538922	Adventure	0.722705	Drama	0.723944	Comedy

Fig. 9 F1 scores of each algorithm on each genre after Gridsearch

Fig. 8 and Fig. 9 show the F1 scores of each algorithm on each genre, after Gridsearch. "x_x" refers to the Action genre, "y_x" refers to the Adventure genre, "x_y" refers to the Drama genre, and "y_y" refers to the Comedy genre. F1 scores were used to compare each of the algorithms. We found that the Neural Network algorithm produced the best results when it had larger amounts of data.

V. Related Work

A. *Classification Using Weighted Kernel Logistic Regression*

Weighted kernel logistic regression was used to classify video genres in [7]. The goal of the authors was to find a way to The authors achieved fast and accurate results. The accuracy was 88% after applying a 10 fold cross validation process with a max number of 30 iterations. The method is different from ours in that the authors use first shot detection to extract video information from videos on sites like Youtube and Youku. The data was then normalized and fed into a Weighted Kernel Logistic Regression model, and then trained and predicted. At 88% accuracy, the model seems to do better than ours tests, but ours is not far behind in accuracy. Our model is more practical and easy to run, as it does not require any manual gathering of first shots. Also, both models serve separate use cases, as one is derived from video sampling and the other from user feedback.

B. *Genre Based Movie Recommendation System*

The authors developed a more targeted movie recommendation algorithm based on the genre of the movie [8]. They used rated scores for genres to classify movies into genre clusters. Then, when a request for recommendation is made, the algorithm computes the genre preference of the user and similar genres, and output a recommendation list. The algorithm yielded slightly more better predictions than those of existing movie recommendation systems. The problem studied in [8] is different because the focus is on movie recommendation instead of genre prediction. They use genres that are already calculated and base their calculations off of that. Our prediction algorithm can enhance the results in [8] as it can cluster movies into more categories based on similarity.

VI. Conclusion

Overall, our classification system performed well using the machine learning algorithms. We were able to get F1 scores of about 0.8 on average for the algorithms on each genre. We found that the Neural Network got the best results for algorithms with more data. Logistic Regression seemed to perform best in most cases, regardless of the amount of data tested. The Action genre was classified most accurately for each algorithm, and Drama consistently performed worse than the other genres.

VII. Work Division

Every single group member worked on each part of the project. All group members participated in finding the datasets and choosing the project type. We also spent time analyzing and figuring out what we could do with the data we had. Each of us contributed to the code and algorithm implementation, although Gadfrey and Michael focused most on it. Each of us

also contributed to the report, although Paul focused most on it. The presentation was done collaboratively. Overall, each group member was involved in each area of the project.

VIII. Learning Experience

First and foremost, this was our first experience with writing a research paper in the IEEE format. We learned about all of the different factors that go into it, the formatting, and the most effective ways to present data. We also learned to utilize plotting to get charts about the data we received in Pandas and Sklearn. Since this project was an accumulation of the two previous projects, it has helped us to solidify the concepts that we were introduced to earlier and to make processing more efficient when working with large amounts of data. We also learned about the strengths and weaknesses of each of the machine learning algorithms. For example, SVM does not work well with problems that are not binary, and the deep learning algorithm requires a large amount of data to get a high F1 score. This project has also improved our ability to increase accuracy with optimizations like parameter tuning and grid search. Finally, this project has given us the full stack experience of going from start to finish. We started with raw data and were able to draw conclusions and analyze the results that we received. It has been a helpful learning experience for each of us.

References

[1] (2017) ML Cheatsheat. [Online]. Available: http://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

[2] R. Saxena. (2017) How Decision Tree Algorithm Works on Dataaspirant. [Online]. Available: http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

[3] H. Chen, 'Data Mining Lecture 4', CSU Sacramento, 2018.

[4] H. Chen, 'Data Mining Lecture 6' CSU Sacramento, 2018.

[5] H. Chen, 'Data Mining Lecture 5' CSU Sacramento, 2018.

[6] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (Tiis) 5, 4: 19: 119:19. https://doi.org/10.1145/2827872

[7] A. Hamed, R. Li, Z. Xiaoming, and C. Xu, "Video genre classification using weighted Kernel Logistic Regression," *Advances in Multimedia*, 2013, pp. 1-6, 2013.

[8] T. Hwang, C. Park, J. Hong, and S. Kim, "An algorithm for movie classification and recommendation using genre correlation," *Multimedia Tools and Applications*, vol. 75:20, pp. 12843-12858, October 2016