# ML1819 Research Assignment 2

### **Team ID:** 19

### **Task ID and Title:** 102. Dataset Pruning: What is the effect on Machine Learning Performance?

### **Student Names and IDs:**

### Aneek Barman Roy - 18304921

### Debrup Chakraborty - 18304460

### Viren Chhabria - 18301780

### **Contribution:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Student ID | GitHub username | Contribution | Percentage |
| Aneek Barman Roy | 18304921 | iamaneek | * Generated visualizations using python * Implementation of Ridge Regression * Literature Review * Report content writing | 33.33% |
| Debrup Chakraborty | 18304460 | rupdeb | * Generating visualizations from Excel * Implementation of Support Vector Regressor * Parameter tuning for the algorithms * Literature Review * Report content writing | 33.33% |
| Viren Chhabria | 18301780 | chhabriv | * Data Pre-Processing * Implementation of Random Forest Regressor * Comparison of evaluation metrics * Literature Review * Report content writing | 33.33% |

1. **Word Count:**
2. **Exceptional circumstances:** N/A
3. **Repository link:** <https://github.com/chhabriv/ML1819--task-102--team-19>

### **Repository contributor’s link:** <https://github.com/chhabriv/ML1819--task-102--team-19/graphs/contributors>

### **Commit Activity:**

Impact of Data Pruning on Machine Learning Algorithm Performance

|  |  |  |
| --- | --- | --- |
| Debrup Chakraborty School of Computer Science & Statistics Trinity College Dublin Dublin, Ireland chakrabd@tcd.ie | Viren Chhabria School of Computer Science & Statistics Trinity College Dublin Dublin, Ireland chhabriv@tcd.ie | Aneek Barman Roy School of Computer Science & Statistics Trinity College Dublin Dublin, Ireland barmanra@tcd.ie |

# *Abstract:* Dataset pruning is the process of removing sub-optimal tuples from a dataset to improve the learning of a machine learning model. In this paper, we compared the performance of different algorithms, first on an unpruned dataset and then on an iteratively pruned dataset. The goal was to understand whether an algorithm (say A) on an unpruned dataset performs better than another algorithm (say B), will algorithm B perform better on the pruned data or vice-versa. The dataset chosen for our analysis consisted of the user ratings of 10,841 applications from the Google play store. The ratings were in the range of 1-5. The dataset was pruned iteratively based on the number of reviews and three regression algorithms to predict the application ratings were implemented. The results indicated that an algorithm that performed better on an unpruned dataset also performed better on a pruned dataset.

***Keywords***: data pruning

# Introduction

A fine line separates cleaning and pruning of a dataset. Cleaning mostly is a preprocessing step that involves removing unrequired data, data imputation, standardizing or normalizing the feature ranges and converting categorical values to numbers [2] [3]. In comparison pruning takes place after preprocessing, where certain data is strategically removed to improve the machine learning model. In this paper we try to bring forth the effect of dataset pruning on the performance of different machine learning algorithms, i.e. If an algorithm (say A) on an unpruned dataset performs better than another algorithm (say B), will algorithm B perform better on the pruned data or vice-versa.

# RELATED WORK

Data pruning had been defined in 2005 as an automated process of noise cleaning and the performance of this mechanism was measured using SVC and AdaBoost algorithms [4]. Removal of certain portions of the dataset is determined to be worthwhile and said to affect the performance of machine learning algorithms [4].

A mathematical model was proposed to predict the success of upcoming movies based on correlation of factors affecting the success of a movie [5].

Automatic rating prediction was proposed in 2011 using the IMDb dataset, however the results were inferior to baseline which was attributed to the dataset lacking diversity in terms of user rating [6].

# METHODOLOGY

## Dataset

The dataset chosen is from the largest Android application store, Google play store <INSERT REF>. It contains 10,481 applications (app) with 13 attributes, where “Rating” indicates the app ratings on a scale of 1-5. The histogram in Figure 1 shows the frequency distribution of the app ratings depicting the rating between <RANGE HERE> to be the highest. The different features in the dataset along with their datatype are shown in <TABLE REFERENCE>

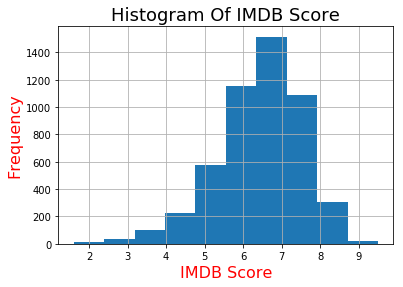


Figure 1: Frequency of IMDb Score of raw dataset

## Pre-processing

Google Play app ratings have continuous values in the range 1-5. The percentage of missing data per column in the dataset can be seen in <TABLE HERE>. The ratings column, which is the target variable had 13.6% of the data missing. 5 other features also had missing data, but the number was less than 1%. Consequently, all the rows containing missing data were removed, since it was not reasonable to impute the target variable and then use it for building the model <SEARCH REFERENCE>. Categorical data in the numeric columns were replaced with -1. Duplicate tuples were removed. Categorical features was transformed to numbers using LabelEncoder and OneHotEncoder. The feature data was standardized using StandardScaler. The aforementioned utilities were used from the scikit-learn library [7].

Heatmap shown in Figure 2 represents the correlation between the features in the dataset. The independent variable ‘Genre’ was dropped from the dataset since it was highly correlated with the variable ‘Category’.

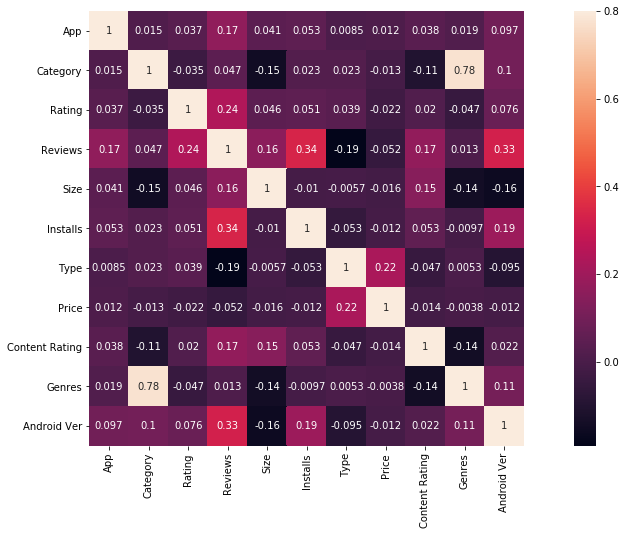


Figure 2: Correlation between the selected features

Random Forest was used to find the most important features in the dataset. The variable importance plot can be seen in <FIGURE REFERENCE>

## Data Pruning

The pre-processed dataset was then pruned based on the number of user reviews for an app. The dataset was iteratively pruned where an app had received less than {1...20} user reviews. This was done as a lower review count would make the rating of that app biased to a small (<20) number of user opinions. Figure 3 shows a scatter plot depicting the number of user reviews v/s app rating.

<FIGURE HERE>

Figure 3: Plot of IMDB score v/s number of users who reviewed on a raw dataset

## Algorithms

Linear Regression, Random Forest Regression and Support Vector Regression were used to evaluate the sensitivity of machine learning algorithms to data pruning. All the algorithms were implemented using the Scikit Learn framework <INSERT VERSION AND REFERENCE HERE>. Hyper parameter tuning for these algorithms was done while training the model using the unpruned dataset, and the best parameter measure from the findings were used for each iteration of the pruned dataset. K-fold cross validation, part of the SciKit Learn Framework was used to find the best parameter values <INSERT REFERENCE HERE>. The value of k used was for all the cases discussed in this paper <FIND REFERENCE FOR k=10>.

*3.4.1 Linear Regression:* This was the simplest algorithm used for the prediction of the ratings. No hyper parameters were used for this algorithm.

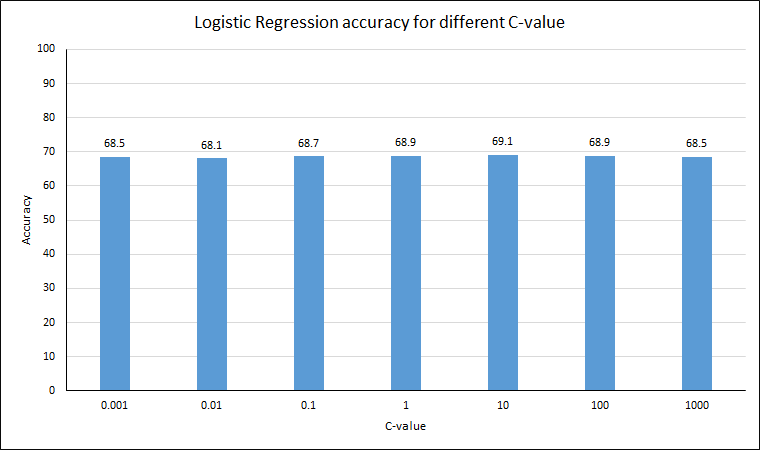


Figure 4: c-value accuracy of logistic regression

*3.4.2 Random Forest Regression:* The n-estimator (number of decision tree classifiers) for random forest was experimented with in the range 10 to 100 in increments of 10, and it was found to be best at <VALUE> as shown in Figure 5.

<FIGURE HERE>

Figure 5: n-Estimator accuracy for random forest

*3.4.3 Support Vector Machine (SVM):* SVM weights were tuned to prevent overfitting on larger margins. For c (regularization parameter) in the range [0.001, 0.01, 0.1, 10, 25, 50, 1000], SVM was found to perform best for c =25 as shown in Figure 6.

<FIGURE HERE>

Figure 6: c-value accuracy of SVM

## Evaluation

10% of the dataset was kept aside at the beginning. This was done to mimic this data as the future/out-of-sample data to test the performance of the model. 90% of the dataset was used to create the model and find mean validation metrics using k-fold cross validation with k=10. The MSE, RMSE and R2 metrics for cross validation vs percentage of data pruned for each algorithm can be seen in <FIGURE HERE>.

*<FIGURE>*

Figure 7: Accuracy of train-test dataset split for various algorithms

The same metrics on the out-of-sample data vs percentage of data pruned for each algorithm can be seen in <FIGURE HERE>

<FIGURE>

# RESULTS AND DISCUSSION

## Metrics

The models have various predictive powers which needs proper measures to evaluate the classifier. We have used accuracy score and F1-score as the evaluation metrics [8].

*4.2.1 Accuracy Score:* A common metric which is the fraction of the samples correctly predicted. For a predicted value of ith sample i.e.   and  being the respective true value, the fraction of right predictions over   may be defined as:

The mean and standard deviation of the accuracy of the three algorithms has been shown in Table 1.

Table 1: Mean and deviations of accuracies

|  |  |  |  |
| --- | --- | --- | --- |
|  | Random Forest | Logistic Regression | SVC |
| Mean (%) | 71.99 | 66.79 | 64.86 |
| Standard Deviation | 1.55 | 1.39 | 1.46 |

Result for each iteration: Table 3, Figure 8

*4.2.2 F1-Score:* We selected this metric to strike a balance between precision and recall. For =1, F1 is derived from:

The mean and standard deviation of F1 scores for the three algorithms have been mentioned in Table 2.

**Table 2**: Mean and deviations of F1 score

|  |  |  |  |
| --- | --- | --- | --- |
|  | Random Forest | Logistic Regression | SVC |
| Mean | 0,70 | 0.63 | 0.58 |
| Standard Deviation | 0.02 | 0.02 | 0.02 |

Result for each iteration: Table 4, Figure 9

## Discussion

We started with an unpruned dataset and then ran 20 iterations to prune the dataset to check how the three algorithms performed with each iteration. For the 0th iteration, the dataset was unpruned and random forest classifier performed the best as shown in Figure 8 and Figure 9. The accuracy score and F1 score fluctuated as per Table 1 and Table 2 with each iteration, but the ranking of the algorithms remained unchanged.

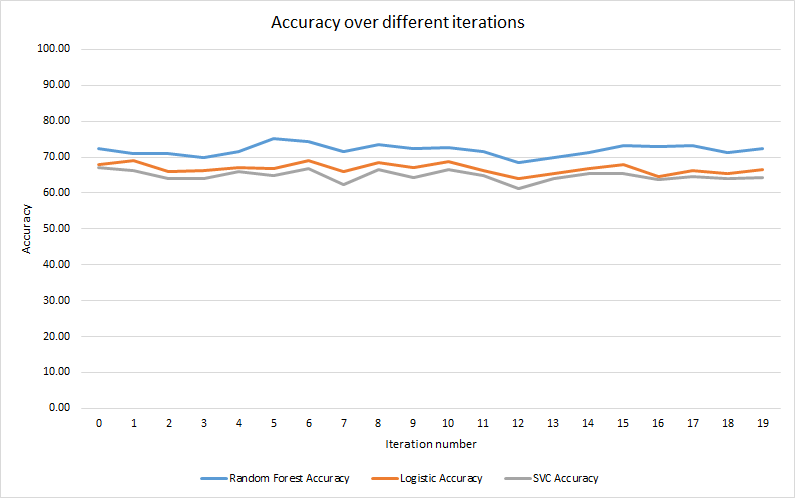


Figure 8: Accuracy score of each algorithm per iteration

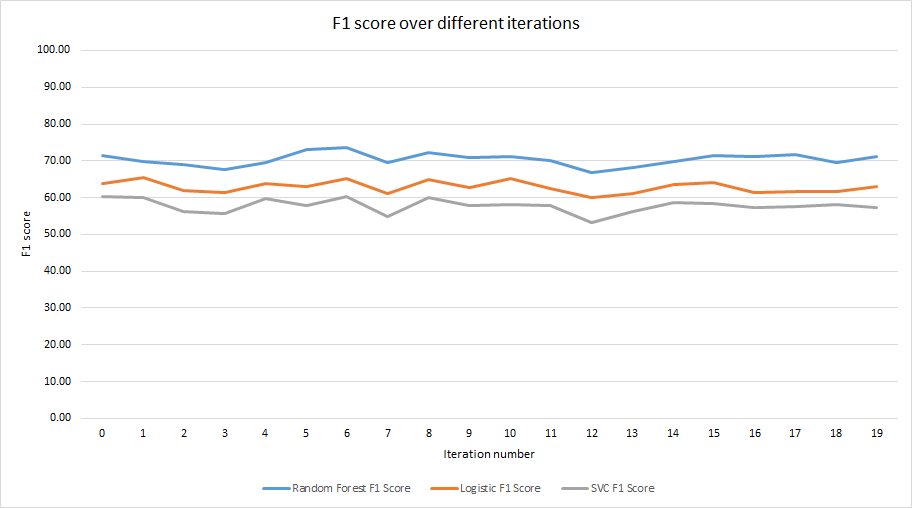


Figure 9: F1 score of each algorithm per iteration

## Results

Some related works on movie datasets were mostly centered on regression trees while some focused improving SVM accuracy [6] [9]. We ran an unbiased analysis on the three algorithms and observed that random forest performed the best followed by logistic regression and SVC as shown in Figure 8 and Figure 9. Their rankings remain unchanged on unpruned and pruned datasets across the two metrics used. However, several iterations showed some fluctuations in their performance. To conclude, pruning of datasets didn’t affect the algorithm performance rankings.

# Limitation and outlook

The dataset had 5043 data points. The limitation of the dataset was that the classes were not evenly distributed among each class of the target variable as shown in Figure 10. This could result in some class of the data being left out of the train/test set. Future work could include using k-fold cross validation to split the dataset. The work can also be improved by confirming the analysis on a different dataset.

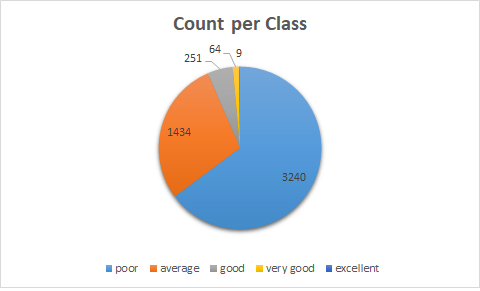


Figure 10: Distribution of movies in each class

# ACKNOWLEDGMENT

This work was conducted as part of 2018/19 Machine Learning module CS7CS4/CS4404 at Trinity College Dublin [10].

# References

|  |  |
| --- | --- |
| [1] | P.-Y. Hsu, Y.-H. Shen and X.-A. Xie, "LNCS 8818 - Predicting Movies User Ratings with Imdb Attributes," *LNAI,* vol. 8818, pp. 444-453, 2014. |
| [2] | "Data Cleaning 101 – Towards Data Science," [Online]. Available: https://towardsdatascience.com/data-cleaning-101-948d22a92e4. |
| [3] | "How to One Hot Encode Sequence Data in Python," [Online]. Available: https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/. |
| [4] | A. Angelova, Y. Abu-Mostafa and P. Perona, "Pruning training sets for learning of object categories," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005. |
| [5] | J. Ahmad, P. Duraisamy, A. Yousef and B. Buckles, "Movie success prediction using data mining," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017. |
| [6] | P. Biljanović and E. a. M.-M. Croatian Society for Information and Communication Technology, MIPRO 2011 : 34th International Convention on Information and Communication Technology, Electronics and Microelectronics : May 23-27, 2011, Opatija, Croatia : proceedings, Croatian Society for Information and Communication Technology, Electronics and Microelectronics, 2011. |
| [7] | "4.3. Preprocessing data — scikit-learn 0.20.0 documentation," [Online]. Available: http://scikit-learn.org/stable/modules/preprocessing.html. |
| [8] | "Model evaluation: quantifying the quality of predictions," [Online]. Available: http://scikit-learn.org/stable/modules/model\_evaluation.html#accuracy-score. |
| [9] | V. Subramaniyaswamy, M. V. Vaibhav, R. V. Prasad and R. Logesh, "Predicting movie box office success using multiple regression and SVM," in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017. |
| [10] | Joeran Beel and Douglas Leith. Machine Learning (CS7CS4/CS4404). Trinity College Dublin, School of Computer Science and Statistics. 2018. |

**Appendix**

**Table 3**: Comparison of Accuracy score of Algorithms (%)

|  |  |  |
| --- | --- | --- |
| Random Forest | Logistic Regression | SVC |
| 72.50  71.01  71.01  69.81  71.50  75.21  74.35  71.53  73.48  72.42  72.56  71.46  68.48  69.96  71.23  73.24  72.91  73.25  71.38  72.50 | 68.00  68.99  66.12  66.33  67.18  66.84  69.16  65.90  68.55  67.05  68.64  66.24  64.10  65.45  66.70  68.04  64.53  66.16  65.35  66.56 | 67.00  66.36  64.09  64.07  66.05  64.98  66.77  62.46  66.67  64.42  66.63  64.86  61.22  64.16  65.30  65.55  63.66  64.63  64.04  64.25 |

**Table 4**: Comparison of F1 score of Algorithms

|  |  |  |
| --- | --- | --- |
| Random Forest | Logistic Regression | SVC |
| 0.71  0.70  0.69  0.68  0.70  0.73  0.74  0.70  0.72  0.71  0.71  0.70  0.67  0.68  0.70  0.71  0.71  0.72  0.69  0.71 | 0.64  0.65  0.62  0.61  0.64  0.63  0.65  0.61  0.65  0.63  0.65  0.63  0.60  0.61  0.64  0.64  0.61  0.62  0.62  0.63 | 0.60  0.60  0.56  0.56  0.60  0.58  0.60  0.55  0.60  0.58  0.58  0.58  0.53  0.56  0.59  0.59  0.57  0.58  0.58  0.57 |