Classifying Software Pirates in the Music Production Software Industry

1. Introduction

This project attempts to dive deeper into the dataset used for the report "The Pricing of Digital Goods in the Music Production Software Industry" [1] to try classify people into those who have pirated music production software and to those who have not based on a variety of features. This could then be used to explore the factors driving people into software piracy to gain more insight into this prominent modern phenomenon that extends to all online markets. This information can unlock economic insights into people's online behavior and help software companies maximize their profits by conducting appropriate customer segmentation, which would likely benefit the customers as well in situations where they have not previously been able to afford the products.

Section 2 will provide an overview of the problem formulation and section 3 will present the dataset and the features used, as well as justify the selection of the used machine learning methods. Section 4 will discuss the results of their application, trying to answer the questions of which factors most correlate with software piracy and which model would be preferred in a classification task.

2. Problem formulation

The type of classification in question is supervised binary classification using multidimensional data points that each represent an individual's responses to all 24 questions of the survey and contain both categorical and continuous data about the respondent's demographics, preferences and sentiments. A survey respondent will be classified as a software pirate if they have answered with a non-zero value to a question about the percentage of pirated software in their collection and as a non-pirate otherwise. The labels are thus the answers to this question, binarized to 1s and 0s.

3. Methods

3.1. Dataset

The dataset was gathered in the summer of 2021 via an online survey distributed through tens of music production related discord servers, subreddits and Facebook groups utilizing convenience sampling. It was also sent to musical institutions and online schools, finding its way to a far-reaching newsletter as well. In total, there were 553 respondents from around the world, but with emphasis on western countries. The respondents were of varying levels of expertise and age. The service hosting the survey was surveyplanet, from where the data was downloaded in csv format.

One datapoint (row) represents one respondent's answers to all 24 questions of the survey, the types of which varied widely. Categorical questions were used to assess the demographics and consumer behavior of the respondents whereas continuous numerical data were collected about sentiments and free-form text-fields were provided for further mapping out of the industry and sentiments of the consumers.

3.2. Features

The raw dataset was data pre-processed by formatting and cleaning up answers and combining columns. The labels were binarized from the 'Percentage of Pirated Software' column and some feature engineering was conducted in a similar manner for 'Can Afford Everything' and 'Ease of Pirating is Important'. Categorical data was converted into dummy variables using one-hot encoding and the respondents' countries of residence were compiled into more general regions such as North America, Western Europe, Northern Europe etc. to have larger sample sizes in each – now more even – category and to reduce the dimensions of the feature space when encoding the nominal categories using one-hot encoding.

Features were selected based on their meaningfulness to the research question of which factors drive people into software piracy with a combination of domain specific knowledge and visual inspection. They are mostly demographic and similar data selected based on where differences were observed between the nearly perfectly split halves of pirates and non-pirates by overlaying differently colored histograms on top of each other for each. Due to their demographic nature, they are easy to observe or estimate as opposed to, for example, the questions on ideal prices by product type and therefore they are informative for, for example, market segmentation and analysis.

The features selected are: Expertise, Goal, Age Group, Region, Importance of ease of pirating when considering a purchase and whether the person can afford everything they need, all of which were one-hot encoded or otherwise binarized to make up a total of 35 binary features. Ease of pirating was predictably the only graph, clearly distinguishing between pirates and non-pirates with zero noise and is therefore perhaps the most obvious feature. Being able to afford everything is yet another feature, with which inspecting the graph shows a clear disparity between pirates and non-pirates. Region and age are again features, which show differences by margins of often over two thirds, which is to be expected as wealthier regions and older people generally have more disposable income. Expertise and goal are also demographic features exhibiting slight differences between pirates and non-pirates.

The dataset is divided into 80% training set and 20% test set, where the 80% is used for training, validating and thus fine-tuning the models via stratified K-fold cross validation with $k = 5$ to maximize diversity, representation and usage of very limited data in training. Once the hyperparameters maximizing the accuracy of the models are selected, this whole 80% is used for training the final model that is then tested with the novel 20% that the models have not seen at any point yet. This 80/20 split is frequently discussed [2] but it is also supported by Google's Pre-ML checklist [3] which recommends the number of examples to be ten times the number of features. However, due to very noisy data, a larger test set is preferable. Therefore, using K-fold cross validation with $k = 5$ and 35 features per data point hits the sweet spot as $553 * 0.8^2 = 353.92$ which is greater than $35 * 10 = 350$, while the size of the test set is maximized.

3.3. Machine learning methods

The first model was a Decision Tree Classifier, chosen due to its high interpretability, which is excellent for the research question of which factors correlate with piracy. It is also great for binary classification with multiple (one-hot encoded) binary variables and for this task particularly due to its simplicity and resulting low training times, which made it possible to iteratively train it for 567 times to optimize both the maximum depth of the tree as well as the features used for training it. The

default loss function for decision trees in sklearn is Gini impurity, which was utilized for its suitability for binary classification, convenience and fast iteration thanks to its efficiency and scalability.

The second model was Logistic Regression as it can also perform feature selection with L1 penalty and it is quite simple and fast as well and thus works as a great benchmark for many binary classification tasks. It also outputs probabilities, which is useful for the actual classification task as it quantifies the uncertainty associated with each prediction. The default loss function, logistic loss, was used due to the probability interpretation as well as the convenience it provides.

Validation error was calculated for both with mean accuracies over the K-fold cross validation iterations to connect the hyperparameters more closely to the desired final outcome.

To maximize the accuracies of the models and to answer the research question, in addition to hyperparameter tuning, automatic feature selection was performed for both models. For the Decision Tree Classifier, this meant training it with every combination of the feature categories and picking the one that maximized accuracy, whereas the Logistic Regression classifier did this automatically by adjusting its weights with the L1 penalties, which correspond to the directions and magnitudes of the effects of each individual feature.

4. Results

The categories of features selected for the Decision Tree Classifier were expertise, age group, region and importance of ease of pirating, which were organized into the tree as shown in figure 1, whereas the Logistic Regression classifier emphasized the importance of ease of pirating as the most important feature positively contributing to increased likelihood of piracy. The other positively contributing factors were being between 0 – 17 or 18 – 25 years old, whereas the negatively contributing factors in decreasing order of influence were being from North America, being able to afford everything, being from Western Europe and being a hobbyist. Based on this, it can be concluded that the two most important factors determining or correlating with piracy in this very limited dataset are age group and region.

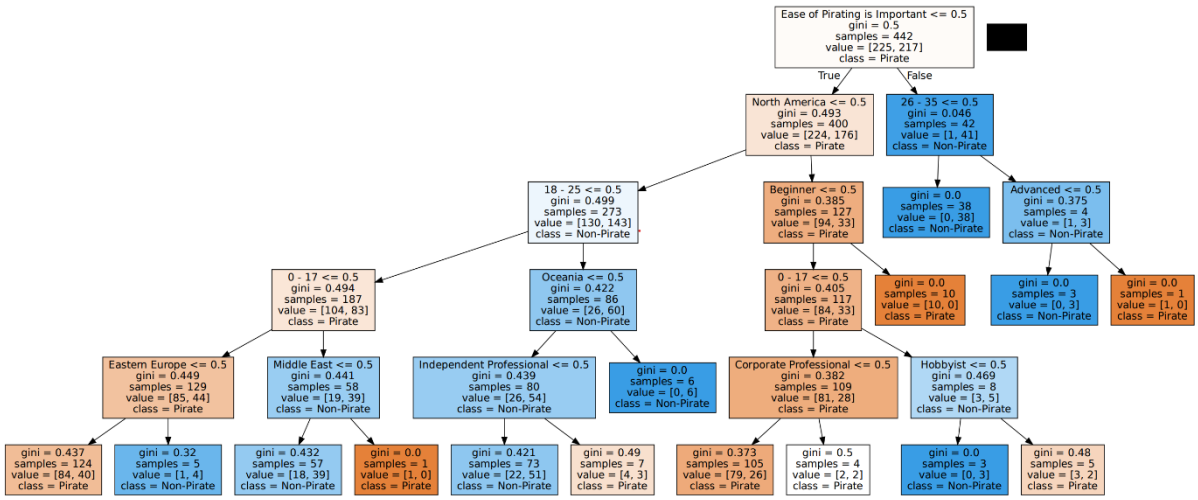

Figure 1 – The final decision tree model

As to the performance characteristics of the two models, they are so close that they are practically identical as can be seen in table 1. The Decision Tree Classifier has classified only one more pirate correctly, giving it the ever-so-slight edge in all the metrics. In fact, this is likely caused by random

chance as altering the random states of the models produced greater variation than this result portrays between the two. If only comparing these numbers, one would have to pick the decision tree, which scores only marginally higher on all metrics, but as this was most likely due to mere chance where random_state = 100 just happened to favor it, other characteristics of the models must also be considered.

	Accuracy	*Precision*	*Recall*	*F1 Score*
DecisionTreeClassifier	0.738738	0.676923	0.846153	0.752136
LogisticRegression	0.729729	0.671875	0.826923	0.741379

Table 1 – Evaluation metrics of both machine learning models

Interestingly, the validation accuracies peaked at more noticeably different values where the DecisionTreeClassifier did not go above 0.70, whereas LogisticRegression reached almost 0.73 at its peak. Even though LogisticRegression managed to correctly classify one less pirate in the end, its validation error was lower and it may be of interest to consider its interpretability in the context of the research question of which factors most correlate with piracy. The decision tree implies these when read from top to bottom and when considering which feature groups were selected but the logistic regression classifier expresses these explicitly through its scaled weights, that communicate both the magnitudes as well as the directions of the relative influences for each individual feature as seen in figure 2, making it the superior model for this project. Its test error is the inverse of its accuracy $1 - 0.729729 = 0.270271$ and it was calculated using the remaining 20% of the data, the splitting of which was already discussed in section 3.2.

```
[('North America', -0.7774139414801525),
 ('Can Afford Everything', -0.4830252662276539),
 ('Western Europe', -0.18415729455530533),
 ('Hobbyist', -0.05119032120880527),
 ('0 - 17', 0.5495077530513445),
 ('18 - 25', 0.6415058922810647),
 ('Ease of Pirating is Important', 1.5628589424119608)]
```

Figure 2 – LogisticRegression weights

5. Conclusion

Two machine learning models, DecisionTreeClassifier and LogisticRegression were developed to classify software pirates using demographic and similar, one-hot encoded, categorical data. Their performance characteristics were practically identical and thus LogisticRegression was selected due to its better interpretability, which poses that the factors most correlating with online piracy are its ease and the age and residence region of the person, both of which usually directly affect their disposable income. This implies that there might still be more room for further market segmentation in the form of, for example, country-specific pricing and student discounts.

The selected Logistic Regression model has an accuracy of 0.729729 and an F1 Score of 0.741379, which is quite good for a dataset this small, biased and noisy. This was enough to reveal and rank overall trends in terms of their approximate influence on the amount of piracy, but the accuracy would be quite poor for a classification system that would mislabel over one fourth of the people considered, which is something to be very careful about. Hence, this project's focus on the predictive features over the predictions themselves.

Potential improvements include, of course, getting more data with more even distributions of features, which would decrease bias and noise. More varied data, for example, in terms of demographics would also enable new insights, whereas with this particular dataset, more features could be used, incorporating continuous data to the classification models as well but this would likely reduce the models' usability in the real world as that is more difficult to acquire and less actionable.

6. Bibliography

[1] M. Keimiöniemi, “The Pricing of Digital Goods in the Music Production Software Industry,” Oct. 2021. Accessed: Sep. 22, 2023. [Online]. Available: https://users.aalto.fi/~keimiom1/portfolio/writing/the-pricing-of-digital-goods-in-the-music-production-software-industry.html

[2] B. Allison, “Answer to ‘Is there a rule-of-thumb for how to divide a dataset into training and validation sets?,’” Stack Overflow. Accessed: Oct. 11, 2023. [Online]. Available: https://stackoverflow.com/a/13623707

[3] “Is My Data Any Good? A Pre-ML Checklist.” Google Cloud, 2018. Accessed: Sep. 22, 2023. [Online]. Available: https://services.google.com/fh/files/blogs/data-prep-checklist-ml-bd-wp-v2.pdf

7. Appendix