Predicting different legs in Toontown Rewritten with supervised classification

Ethan CZ

March 2, 2022

Abstract

In this project, three supervised classification algorithms, namely a logistic regression, random forest classifier, and support vector machine, were trained on a preexisting dataset consisting of various features of 3,000 Toons from the MMORPG Toontown Rewritten with the goal of predicting whether or not Toons in a previously unseen dataset compiled for the purpose of the task had legs which were a different colour than the rest of their body. Results from a comprehensive study performed on the preexisting dataset were used to inform the current task. None of the algorithms were able to accomplish their objective, with all three models returning very low F_1 scores for the presence of the target feature. This outcome was likely prompted by the feature's lack of systematicity and the relevance of personal and aesthetic taste in its presence or absence. For these reasons, future machine learning activity in this specific area may be challenging, although machine learning analyses are envisioned for various other Toon features which have the potential to be more methodical, such as interactions of missing and organic Gag tracks.

Contents

1 Introduction

1.1 Background

Machine learning is a rapidly expanding domain of computer science, or, more specifically, artificial intelligence, dealing with algorithms and systems which can improve and learn from experience. Although certain machine learning algorithms possess parameters which can or must be set, the algorithm builds its own model and produces output without human direction based on the data provided to it. Many varieties of machine learning algorithms exist, of which the main two are supervised and unsupervised learning algorithms. Supervised learning algorithms build a model based on training data consisting of both inputs and desired outputs, and unsupervised learning algorithms attempt to find structure in a dataset based solely upon unlabeled inputs. Machine learning algorithms also aim to accomplish either classification, where the model identifies which of a set of categories an observation belongs to, or regression, where the model outputs a continuous numeric value. Classification yields a discrete label while regression yields a quantity. The task at hand is a supervised classification problem whose goal is to predict whether or not an observation possesses a certain feature.

1.2 What is Toontown Rewritten?

Toontown Rewritten, released in 2013 as a reincarnation of Disney's Toontown Online created by the game's community, is an online game in which the player creates a character known as a Toon and completes tasks in order to increase their maximum health points, or Laff, and advance through a series of in-game areas. Toons are highly customizable, and the player is able to choose from various options for species, colours, clothing items, name tag styles, and weapons, known in the game as Gags. There are eight different varieties of Gags, called Gag tracks, and a Toon can possess seven in total. The player can also opt to grow one of their Gag tracks in their garden, after which it is referred to as organic and becomes more powerful, increasing in either accuracy, damage, or healing power.

1.3 Purpose

The purpose of this analysis was to determine to what degree a supervised classification algorithm could determine whether or not a Toon had been given legs which are a different colour than the rest of its body, a trend in the game which is somewhat associated with a certain subculture. Each Toon possesses a multitude of characteristics, and this analysis aims to ascertain which of them, if any, are relevant to the presence or absence of the target feature.

1.4 Results

All three classification algorithms implemented to carry out this analysis performed worse on testing data than on training data. On previously unseen testing data, a logistic regression returned an accuracy score of 0.839 and F_1 scores of 0.911 for the absence of the target feature and 0.191 for its presence, a random forest classifier returned an accuracy score of 0.816 and F_1 scores of 0.896 and 0.227, and a support vector machine returned an accuracy score of 0.808 and F_1 scores of 0.892 and 0.143. None of the algorithms were able to demonstrate any meaningful predictive power for this task.

1.5 Significance

This study brings to light the implications of the role of personal choice in variation. Even with strong predictors which had already been proven to be significantly correlated with different legs, all three classification algorithms performed poorly, and very little of the variation was able to be accounted for by the models. In contexts where aesthetic preference plays a large role in the presence or absence of a feature, it is difficult to obtain meaningful or useful results from a systematic algorithm.

2 Problem definition and algorithms

2.1 The task

The task at hand is to train a supervised machine learning model to predict whether or not a Toon has legs which are a different colour than the rest of its body. This will require a classification algorithm whose inputs consist of a subset of the features of the dataset, the members of which are to be determined through exploratory analysis and visualization, and which outputs one of two possible values indicating whether or not the Toon possesses different legs. This is an interesting problem because it is situated at the intersection of preexisting demonstrated statistical correlations and creativity and personal preference. Previous work in this area has established that certain features are significantly more likely than others to co-occur with different legs, but are these correlations powerful and consistent enough for a classification algorithm to use them to accurately predict the feature in question, or is its nature too fundamentally grounded in personal preference for this to be feasible?

2.2 Algorithms

Four algorithms were selected to carry out this task. K-modes clustering was used to assist in exploratory analysis and feature engineering, and logistic regression, random forest classifiers, and support vector machines were selected as classification algorithms.

The k-modes algorithm is an extension of the k-means algorithm[1] which deals with categorical data through matching dissimilarity measures and the use of modes instead of means (Huang, 1998). These dissimilarities are computed by comparing members of the dataset to each other and counting the number of categorical values which differ between them (Cao et al., 2011). An instance of the algorithm is created and given a constant k indicating the number of clusters into which to partition the dataset. k initial modes are selected, one for each cluster, and each member of the dataset is allocated to the cluster whose mode bears the most similarity to its own feature vectors. The mode of each cluster is continuously recalculated and updated as objects are allocated. Dissimilarities are recalculated after the entire dataset has been categorized into clusters, and objects are shuffled accordingly. Clustering is complete when no observation changes clusters when dissimilarities are recalculated across the dataset (Huang, 1998). K-modes clustering was implemented because of its potential to illuminate more subtle correlations between groups of observations, which would not be evident from visualization alone, through analysis of clusters displaying a high proportion of the target feature.

Logistic regression uses a logistic function to model a categorical dependent variable. It is often used to predict a binary dependent variable, but can be extended to handle a dependent variable with more than two possible values. The logistic function outputs a probability value between 0 and 1, and whether that predicted value is closer to 0 or 1 determines which level of the dependent variable is predicted for the observation in question (Joby, 2021).

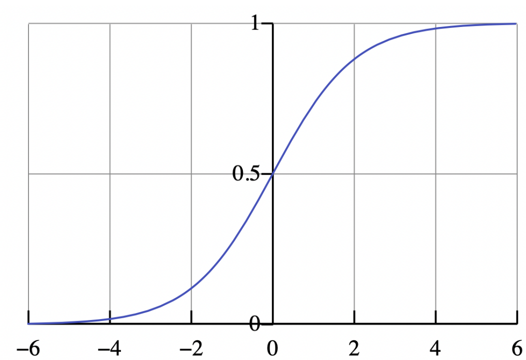

Figure 1: The logistic function takes the shape of a sigmoid curve (Source: Qef, 2014, via Wikimedia Commons).

A binary logistic regression was selected for this

[1] K-means clustering, the algorithm from which the k-modes algorithm is derived, performs the same process as k-modes except it is designed to cluster numerical and continuous data and uses means instead of modes (Huang, 1998).

task because of its simplicity, the nature of the feature at hand, and the algorithm's high degree of interpretability. However, it can overfit on high-dimensional datasets, which is of particular relevance for this task, and it cannot easily be used to represent complex relationships.

Decision trees are a popular machine learning tool which partition the feature space into a tree-like structure, consisting of an initial root node which splits into various internal nodes, terminating in a number of final decision nodes. Each of the internal nodes corresponds to a feature of the dataset, and the node splits into various branches depending on the possible values of that feature. This process continues until a terminal node corresponding to a class of the dependent variable is reached (Sontag, 2012). Decision tree algorithms try to generate an optimal decision tree for a given dataset. A random forest classifier generates a large number of different decision trees, repeatedly classifies a given input using these trees, and outputs the class of the dependent variable selected by the most trees. Random forests were selected due to their ability to smoothly handle heterogeneous data and their reduced tendency to overfit. Complex interactions may exist in a dataset possessing this level of dimensionality, and random forests are able to deal with this. These positive attributes come at a cost of interpretability, as visualizing random forests is much more difficult than visualizing individual decision trees.

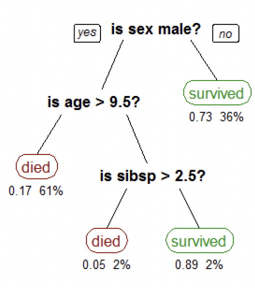

Figure 2: An example of a simple decision tree created for a popular dataset consisting of information about passengers on the Titanic (Source: Milborrow, 2011, via Wikimedia Commons).

Support vector machines perform classification through the creation of one or more hyperplanes which divide the feature space. The observations closest to the hyperplane are known as support vectors, and the objective of the algorithm is to maximize the distance of the hyperplane from these points (Fletcher, 2008).

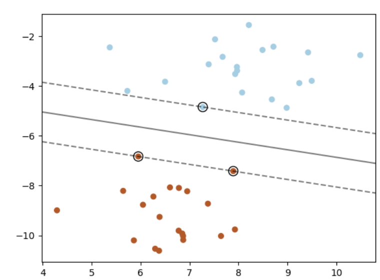

Figure 3: An example of the support vector machine process. The classifier attempts to maximize the distance between the hyperplane and the points closest to it (Source: scikit-learn).

The SVM algorithm was chosen for its ability to scale well to high-dimensional data and its intrinsic tendency to generalize by permitting misclassification, reducing the risk of overfitting, but once again, this model is difficult to visualize in higher dimensions like the task at hand must handle, and tuning the hyperparameters of the model can be challenging and time-consuming.

3 Methodology

3.1 Dataset

Two separate datasets with identical features were used in this analysis. The training data consists of a sample of 3,000 Toons collected from June to August 2021 and the test data consists of a sample of 1,000 Toons collected from January to February 2022. Both datasets were collected by hand and care was taken to record observations as randomly as possible and avoid bias. The data possesses ten features, although `colour`, the Toon's primary colour, and `lc`, the colour of the Toon's legs if it had different legs, were not considered in the analysis at hand.

1. `laff`	Toon's maximum health
2. `species`	Toon's species
3. `gender`	Toon's gender
4. `colour`	Toon's primary colour
5. `dl`	Whether or not Toon's leg colour differs
6. `lc`	Colour of Toon's legs if relevant
7. `mt`	Gag track that Toon does not possess
8. `org`	Toon's organic Gag track
9. `nt`	Toon's style of name tag
10. `flippy`	Whether or not Toon is wearing a Flippy shirt

3.2 Hypotheses

Various observations from previous analytical work on the training dataset were used to inform the hypotheses underlying this task. Toons possessing 120 Laff or more, organic Sound, and certain name tags unlocked later in the game, and missing either the Toon-up, Lure, or Sound Gag tracks were more likely to have different legs (CZ, 2022). The analysis thus began with the inference that these would display predictive power. It was also hypothesized that even the predictors most strongly correlated with the target feature could still perform poorly in a classification task given the role of personal and aesthetic preference in its presence or absence.

3.3 Preprocessing

The training dataset had already been subject to preprocessing from its use in previous analyses. The testing data, which was collected specifically for the purpose of this task, underwent the same preprocessing as the training set. Observations missing information were dropped from the dataset, as well as observations possessing identical values for all ten features, despite the possibility of multiple Toons possessing the same characteristics. An eleventh feature named `range` consisting of eight Laff point ranges, beginning at 60, as the lowest Laff value in the dataset is 62, and increasing in increments of 10 to the maximum Laff value of 139 was also created. This `range` feature was also created in the training dataset during its initial preprocessing[2]. Binary feature vectors, namely `gender`, `dl`, and `flippy`, were encoded numerically.

3.4 Evaluation

Two metrics were used to evaluate the performance of each classifier: the model's overall accuracy and the two F_1 scores, one for each category of the feature at hand. The overall accuracy conveys the fraction of labels in the test set which were correctly predicted by the classification model. The F_1 score is the harmonic mean of precision and recall. Precision refers to the proportion of observations identified as members of a certain category which truly are members of that category, while recall refers to the proportion of members of one category that the classifier is able to correctly identify. Thus, it is possible to obtain a very high precision score but a very poor recall score and vice versa, causing each of these measures to have little explanatory power when taken independently. The F_1 score handles this tug of war. It can be written as follows:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Thus, it is a comprehensive and informative indicator of multiple aspects of a model's overall performance. Both of these metrics were used to judge performance, as opposed to solely taking note of overall accuracy, due to the distribution of the target feature in the training dataset. Only 17.4% ($n = 524$ of $N = 3000$) of the observations in the training dataset possess the target feature, and it would therefore be possible to achieve reasonably high accuracy while catching only a very small portion of Toons with different legs. F_1 scores were taken into account to at-

[2] It is important to note that since the collection of the training data, the maximum Laff that a Toon can possess has increased from 137 to 139. This is why the final bin of `range` in Figure 4 possesses the label 130-137.

tempt to create a model which recognizes as many instances of the target feature as possible.
Cross-validation was performed by repeatedly training and testing new instantiations of each classifier with different splits of the dataset, recording the overall accuracy and F_1 scores of each instantiation, and subsequently taking the mean of each of these metrics. For each potential set of predictors, 30 different training and testing splits were created. A new instantiation of the algorithm was fitted to each training split, predictions were generated, and accuracy and F_1 scores were recorded. The mean of each metric was calculated and used as the overall indicator of the performance of the set of predictors in question, and the set with the highest scores was selected for testing with new data. This selection was not always a straightforward choice, as will be elaborated upon later in this document.

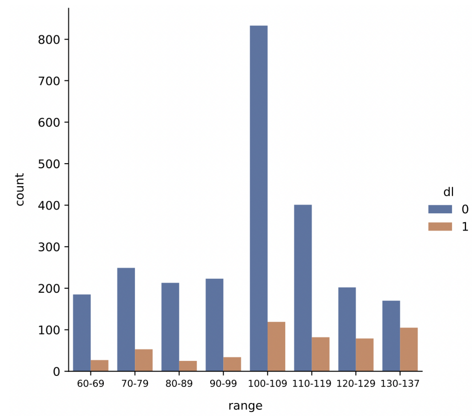

Figure 4: Laff range across Toons with and without different legs (Source: own image).

4 Exploratory analysis

4.1 Visualization

Exploratory data visualization was performed to flesh out and substantiate the preexisting correlations noted in previous work and glean further insight about the distribution of the target feature. Visualization confirmed the preexisting conclusions. Gender also seemed to be a possible predictor, as male and female Toons both showed very similar frequencies of the target feature despite the larger population of female Toons.

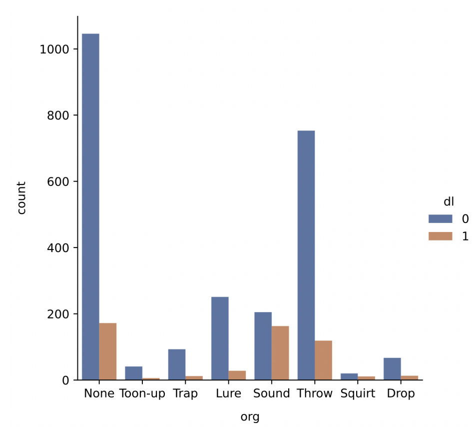

Figure 5: Organic Gag tracks across Toons with and without different legs (Source: own image).

4.2 *K*-modes clustering

K-modes clustering was implemented to explore possible relationships which would not become evident through simple visualizations. An ideal number of clusters into which to segregate the data was determined with an elbow plot. Figure 7 displays the cost, or the sum of the dissimilarities between all of the clusters (Bonthu, 2021), as a function of k from 0 through 50. 40 was selected as the ideal number of clusters for the dataset because cost began to decrease only minimally after approximately this

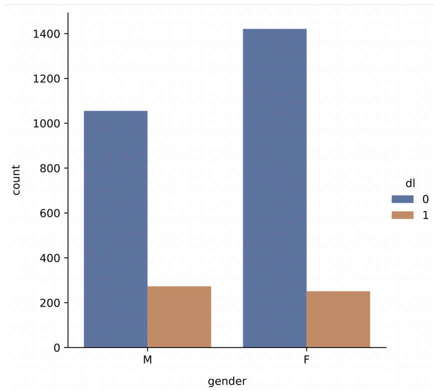

Figure 6: Gender across Toons with and without different legs (Source: own image)

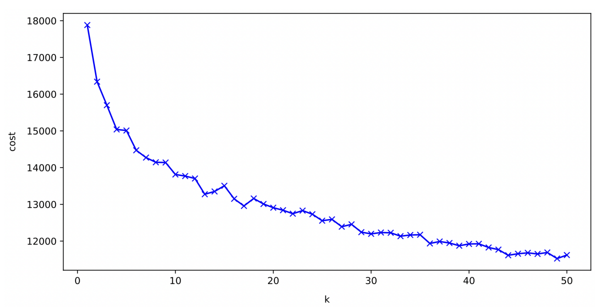

Figure 7: Cost as a function of k (Source: own image).

amount of clusters. The increasing computational complexity of larger values of k was deemed not worth the small decreases in cost. The dataset was subsequently divided into 40 clusters and clusters displaying a large amount of the target feature were aggregated into a separate dataset upon which further exploratory visualization was carried out. Clusters were noticeably black and white in their proportions of different legs, with either nearly all or nearly none of the members in a given cluster displaying the feature.

This new dataset largely mirrored the characteristics of the original dataset in terms of those of Toons possessing the target feature. The only takeaway from further visualization was the large presence of Toons missing Drop among clusters with primarily different legs, whereas it had been originally hypothesized on the basis of previous work that only missing Sound, Toon-up, or Lure were predictors of the target feature.

The `colour` feature was not incorporated into this analysis. Exploratory analysis demonstrated that no colour exhibited a noticeably large amount of the the target feature, and based on that observation combined with the feature's dimensions, as there are 36 colours for which leg colours can be changed, it seemed judicious to exclude it from this analysis. A machine learning context is also simply not how I wish to address a potential relationship between `colour` and `dl`. Future work is planned for those features, but not in this sphere.

4.3 Feature engineering

Based on exploratory analysis, four new features were engineered: `orgsound`, `ab120`, `uncommonmt`, and `highnt`. Organic Sound and possessing 120 Laff or more had already been shown in previous statistical analysis (CZ, 2022) to be significantly correlated with the presence of the target feature, and this was confirmed with further visualization. Drop was added to `uncommonmt` in addition to its original levels of Toon-up, Lure, and Sound based on analysis following k-modes clustering. Figure 8 displays the distribution of different legs across different name tags. The `highnt` feature expands on the initial hypothesis that the Practical and Fancy name tags were predictors of the target feature and includes multiple other name tags which are unlocked later in time, namely Action, Whimsical, Zany, and Triumphant.

5 Results

5.1 Logistic regression

[`flippy`, `orgsound`, `highnt`] performed best, yielding a mean accuracy score of 0.842 and mean F_1 scores of 0.911 for the absence of the target feature and 0.268 for its presence.

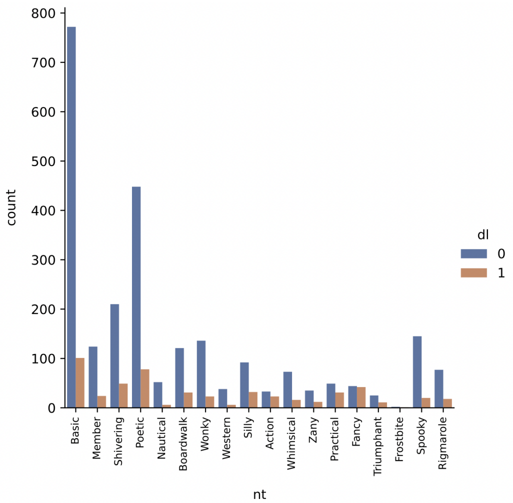

Figure 8: Name tags across Toons with and without different legs (Source: own image).

5.2 Random forest classifier

Dummy variables were generated for each feature which was not numeric or binary. [`laff`, `species`, `gender`, `mt`, `org`, `nt`, `flippy`] performed best, yielding a mean accuracy score of 0.851 and mean F_1 scores of 0.916 for the absence of the target feature and 0.381 for its presence with forests of 200 trees. Random forests outperform logistic regression while not making use of engineered features. The underlying structure of random forests and the decision trees which comprise them allow them to more easily handle larger amounts of features, as well as higher-dimensional features.

5.3 Support vector machines

A radial basis function was used as the kernel, or method of transformation of the data to measure similarity, because of its relative ease of calibration. A grid search was used to compute two hyperparameters for each instantiation, namely `C`, which determines the cost of misclassification, and `gamma`, which determines the sphere of influence of single observations during model training (Yıldırım, 2020). GridSearchCV also confirmed that the radial basis function was the optimal kernel for the task at hand. [`flippy`, `orgsound`, `highnt`] yielded the highest accuracy and F_1 score for predicting the absence of different legs, coming in at 0.840 and 0.910. However, while [`flippy`, `orgsound`, `ab120`] yielded lower values on these two metrics than the aforementioned set of predictors, returning 0.837 and 0.908, its F_1 score for the presence of different legs is much higher, 0.311 in contrast to the first set's 0.226. Because of this discrepancy, the second set of predictors was selected as the best performing set, as the differences in the first two metrics between the groups are extremely small and more instances of the target feature can be caught with the second model.

5.4 Final results

An instantiation of each algorithm was trained on the entire training data set with the determined optimal predictors and subsequently tested on the entire testing data set. A logistic regression returned an accuracy score of 0.839 and F_1 scores of 0.911 and 0.191, the random forest classifier returned an accuracy score of 0.816 and F_1 scores of 0.896 and 0.227, and the support vector machine returned an accuracy score of 0.808 and F_1 scores of 0.892 and 0.143. All three classifiers yielded similar accuracy scores and F_1 scores for the absence of different legs to those obtained during training by the optimal predictors, while F_1 scores for the presence of different legs are all noticeably lower. No algorithm was able to effectively predict the presence of the target feature.

6 Discussion

The final results somewhat mirrored the training outcomes, with the support vector machine returning the lowest accuracy and the random forest classifier returning the highest F_1 score for the presence of different legs. The higher different legs F_1 score yielded by the random forest classifier is to be expected given its underlying structures and its ability to handle high-dimensional datasets more adeptly than a logistic regression, which may fall flat when given a complex relationship to tease apart. The random forest classifier may return a lower accuracy score but a higher F_1 score for the target feature because it is able to begin to tease apart complex relationships and thus correctly select more instances of different legs, but its efforts sometimes lead it astray, leading to misclassification. On the other hand, the logistic regression cannot handle this level of complexity, which works in its favour in terms of accuracy because of the frequency of different legs in the datasets, which will be further discussed later in this section.

The hypotheses put forth prior to executing this task, which were that various features having previously demonstrated correlations with the target feature would wield predictive power in this task but could still perform poorly due to the inherent role of personal and aesthetic choice in the presence of different legs, were correct. `ab120`, `orgsound`, and `highnt` were all engineered as features and were selected as some of the most promising predictors for logistic regression and support vector machines, yet these algorithms performed quite poorly in correctly identifying observations with different legs. Even the strongest predictors captured only a very small amount of variation.

A situation which must be acknowledged in the context of the random forest classifier is the increase in maximum possible Laff points from 137 to 139 during the period between the collection of the training and testing datasets. The random forest classifier thus did not encounter observations possessing values of `laff` which were greater than 137 during training, but did encounter them during testing. This was not relevant for the logistic regression or support vector machine, as neither model dealt with individual values of `laff`, but rather with ranges, and the new values of `laff` both fell into the final bin of the engineered `range` variable. Despite the introduction of these two previously unseen levels of `laff`, the random forest classifier returned the highest F_1 score for the presence of the target feature. This is most likely due to the fact that the two previously unseen values are extremely small increases from values with which the classifier was already very familiar, and the algorithm was able to make inferences regarding these observations based on prior knowledge from members of the training set which had close to this amount of Laff.

Also at play here is the distribution of different legs in the training and testing datasets. Only 17.4% ($n = 524$ of $N = 3000$) of the training dataset and 16.5% ($n = 165$ of $N = 1000$) of the training dataset possess different legs. Because of this distribution, it is possible for a classifier to miss the vast majority of relevant observations and still perform reasonably well. However, even in light of this situation, all three classifiers still detected a very small amount of instances of the target feature.

The majority of the variation in this task is inherently unexplainable due to the nature of the different legs feature. Although there are various characteristics of a Toon which tend to co-occur with it at a statistically significant rate, this is not a systematic feature produced by any sort of underlying process. Personal choice and aesthetic preference bring about the majority of the variation that is observed here, and this sort of variation cannot be untangled by a machine learning algorithm.

7 Future work

My interest in Toontown Rewritten-related machine learning research remains despite the poor results of this task. It is probably most judicious to next tackle an area which arguably is more systematic than aesthetic choices, such as combinations of missing and organic Gag tracks with analysis from the perspective of specific in-game advantages and disadvantages of these combinations.

As mentioned, a machine learning context is not how I envision the exploration of the interaction of colour and leg colour among Toons with different legs. A correlation analysis may be possible for this area, as well as for the analysis of the interaction between colour and the presence of different legs. A much larger corpus of Toons with different legs will be necessary in order to perform work in these areas.

8 Conclusion

The goal of this task was to train a supervised classification algorithm to predict whether or not a Toon had different legs. Unfortunately, none of the three algorithms performed well. A logistic regression and a random forest classifier generated the best results, with accuracy scores and F_1 scores of 0.839, 0.911, and 0.191, and 0.816, 0.896, and 0.227, respectively. The primary takeaway from this undertaking is that there exist kinds of variation which cannot meaningfully be explained or interpreted in a machine learning context. For contexts in which there is no deeper and more systematic algorithm at play than correlations between bundles of features and in which much of the observed variation stems from randomness or personal preference, only so much can be dealt with and predicted by a machine learning model.

Appendices

A Discussion of technical methods

Both datasets were collected by hand in spreadsheets and converted into CSV files. Both raw datasets have been uploaded along with this paper. Analysis of data took place in Jupyter Notebooks using Python. Tidying and manipulation was conducted using pandas and data visualization was performed with matplotlib and Seaborn. The KModes module from the kmodes package was used for k-modes clustering, the LogisticRegression module from sklearn.linear_model was used for performing logistic regression, the DecisionTreeClassifier and RandomForestClassifier modules from sklearn.tree and sklearn.ensemble, respectively, were used to generate random forests, and the SVC and GridSearchCV modules from sklearn.svm and sklearn.model_selection, respectively, were used to implement and tune support vector machines. The classification_report, confusion_matrix, precision_recall_fscore_support, and f1_score modules were imported from sklearn.metrics for scoring and evaluation, and train_test_split from sklearn.model_selection was used to divide the training dataset.

References

Bonthu, H. (2021, June 13). *KModes Clustering Algorithm for Categorical Data.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-\data/

Cao, F., Liang, J., Li, D. et al. (2011). A dissimilarity measure for the k-Modes clustering algorithm. *Knowledge-Based Systems*, *26*, 120-127. https://doi.org/10.1016/j.knosys.2011.07.011

CZ, E. (2022, January 2). *An exploration of Toontown Rewritten demographics.* https://github.com/ethan-cz/data-analytics/blob/main/Toontown-Rewritten-demographics/Toontown%20Rewritten%20demographics.pdf

de Vos, N. (2021). kmodes categorical clustering library. https://github.com/nicodv/kmodes

Fletcher, T. (2008, December 23). *Support Vector Machines Explained.* University College London. https://www.csd.uwo.ca/~xling/cs860/papers/SVM_Explained.pdf

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*, 283-304. https://doi.org/10.1023/A:1009769707641

Hunter, J. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90-95. https://ieeexplore.ieee.org/document/4160265

Joby, A. (2021, July 29). *What Is Logistic Regression? Learn When to Use It.* Learn Hub. https://learn.g2.com/logistic-regression

McKinney, W., & others. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56)

Milborrow, Stephen. (2011, March 1). CART tree titanic survivors.png [online image]. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:CART_tree_titanic_survivors.png

Pedregosa, F., Varoquaux, G., Gramfort, A. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html

Qef. (2014, July 4). Logistic-curve.svg [online image]. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Logistic-curve.svg

Sontag, D. (2012). *Decision trees, Lecture 11.* [Lecture slides]. New York University. https://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture11.pdf

sphx_glr_plot_separating_hyperplane_001.png [online image]. scikit-learn. https://scikit-learn.org/stable/modules/svm.html

Toontown Online [Computer software]. (2013). Burbank: The Walt Disney Company.

Toontown Rewritten [Computer software]. (2021). https://www.toontownrewritten.com

Waskom, M. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*:60, 3021. https://doi.org/10.21105/joss.03021

Yıldırım, S. (2020, May 31). *Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters.* Towards Data Science. https://towardsdatascience.com/hyperparameter\-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167