CS 3840-01

Applied Machine Learning

Dr. Tanvi Banerjee

Final Project Report

Tic-Tac-Toe Dataset

Alex Jervis
Jervis.8@wright.edu

Makhmud Makhmudov
Makhmudov.5@wright.edu

Timothy Stubblefield
Stubblefield.9@wright.edu

April 27th, 2020

Motivation/Goals/Data Description

We thought that with the smaller data set size and universally understood game mechanics, the results would be much easier to analyze, understand, and interpret. The goal of this study is to understand how computer algorithms recognize patterns to properly assess the outcome of a game as simple as Tic-Tac-Toe, where “x” is assumed to have played first. The models are made to recognize one of the 8 possible ways to create a "three-in-a-row” pattern for “x” and accurately decide if “x” wins or doesn’t win.

We believe that our model will more likely predict a positive outcome for “x” rather than a negative outcome. Due to the first player advantage (First Move) the player should “in theory” win more often. With that player advantage, we believe that our model will skew positively & guess more positive results rather than negative. The EDA metrics shown below.

Data Set Characteristics:	Multivariate	Number of Instances:	958	Area:	Game
Attribute Characteristics:	Categorical	Number of Attributes:	9	Date Donated	1991-08-19
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	234280

Algorithm and Evaluation Metrics

When using **kNN** we tried all the different metric types and used 1 neighbor for each. We tried out each **Neural Network**(NN) activation with differing regularization values and hidden nodes and layers. For the **Random Forest** (RF), we used 20 trees and did not split up any subsets. The **Tree** had min leaves 1, max leaves 100, did not split any subsets and stopped at 98%. For the **SVM** model, we found the minimum cost for the **Polynomial** and **RBF** kernels. **Logistic Regression** used both regularization types with C = 200. We included **Naive Bayes** and **Constant** to show models that did not perform very well. As featured below in **Table 1**.

Table 1: Best Performing Models

Algorithm	Model	AUC	CA	F1	Precision	Recall
kNN	Man(1)	1.000	1.000	1.000	1.000	1.000
kNN	Maha(1)	1.000	1.000	1.000	1.000	1.000
kNN	Euc(1)	1.000	1.000	1.000	1.000	1.000
kNN	Cheb(1)	1.000	1.000	1.000	1.000	1.000
NN (α=0.0001)	ReLU (500,1000)	1.000	1.000	1.000	1.000	1.000
NN (α=0.0001)	Tanh (500,1000)	1.000	1.000	1.000	1.000	1.000
RF	20 Trees	1.000	1.000	1.000	1.000	1.000
Tree	Min Leaves = 1 Limit = 100 Stop at 98%	1.000	1.000	1.000	1.000	1.000
SVM Cost = 1.8	Polynomial G = auto C = 25 D = 3	1.000	1.000	1.000	1.000	1.000
SVM Cost =2.77	RBF G = auto	1.000	1.000	1.000	1.000	1.000
NN (α=0.1)	Identity (250x4)	0.998	0.984	0.984	0.985	0.984
NN α=(0.0001)	Logistic (1000,1000)	0.998	0.983	0.983	0.984	0.983
Logistic Regression	Lasso(L1) C = 200	0.998	0.983	0.983	0.984	0.983
Logistic Regression	Ridge(L2) C = 200	0.998	0.983	0.983	0.984	0.983
Naive Bayes	Default	0.769	0.698	0.687	0.686	0.698
Constant	Default	0.500	0.653	0.516	0.427	0.653

Challenges and Solutions

The hardest challenge we faced as a group was communication. Due to the cancellation of in person classes and the move to online classes, all of our lives became a lot busier and it made it much more difficult to connect with each other regularly. We managed to overcome this obstacle by being persistent in our attempts at communicating with one another and finding different ways to make communication more efficient and effective. Another problem we ran into was our original dataset was too large for orange to handle, that on top of the communication issues, made it really difficult to get an early start on the project as we were all on different pages in terms of where we were at with the project.

However, we eventually decided to switch to our current dataset due to its smaller size and clear target and parameters. It made understanding the data a lot easier and helped to get our project back on track. A lot of the models we tried to evaluate with the data set did not work as intended due to the simplicity of the data set. For instance, we received an error notification when trying to create a Linear Regression model and most of the models we tested were able to perform with perfect scoring metrics and minimal complexity in the models. The perfect scoring models also made it difficult to explore more complex model ideas since there weren't really any errors for us to assess how we could improve the models over time when usually the simplest model got the best scores.

Broader Application Area

We were able to successfully prove our hypothesis even though the majority of the models performed with perfect metrics. The models that were not able to obtain perfect metric scores all tended to have a bias of choosing positive when it was negative, which was reflected in their respective recall and precision scores. This was precisely how we predicted the models would perform when the models did not predict correctly for all the test cases.

As for the results, while the kNN was the best performing algorithm, it was also the simplest. Due to kNN working based on the number of neighbors closest to it, by choosing to only use one closest neighbor the model would correctly group all the points, resulting in perfect metrics for all the metric types. Now, for Neural Networks (NN), we could generate flawless metrics with the ReLU function and the tanh function, but were not able to find the correct parameters for the Logistic regression and Identity function to give us similar exemplary metrics for the data. Based on our previous experiences with the kNN and NN models, we decided to expand outside of our comfort zone and test models we weren't familiar with, to see if we were able to achieve similarly perfect results with all the possible models we managed to test. The Random Forest (RF) and Tree models provided perfect scores after minimal parameter manipulation leading us to pursue more complex model types. The Support Vector Machine (SVM) proved very effective across both models, however regardless of the model eventually both of them were able to achieve perfect metrics simply by increasing the cost function value, without changing any other parameter. We included the constant and Naive Bayes models to provide models where because we weren't able to change any of their parameters performed rather poorly compared to the other models.

Future Work/ Next Steps

Due to the lack of time to work with this data set, if given more time, we would likely find parameters for all the models to perfectly assess the data set with perfect metric scores. The greatest limitation when it came to this project was the circumstances surrounding our work, as the act of social distancing created a gap in communication that negatively impacted collaboration and work on this project and our initial dataset not working on orange only delayed our attempts at beginning the first steps of the project.

However, based on the results of this dataset, an idea for future work would be to test these models with other simple game based datasets, the Connect4 dataset being the highest probability if we could figure out how to make it work with Orange. Since these perfect scoring models would probably not recognize other game test states it would be an interesting concept to incorporate both datasets together to make the models more complex. The resulting study would have the goal of trying to uncover how the algorithm handles multiple games with different but similar end states and if there are any similarities in the algorithms made to classify won endgame states.