Using Machine Learning to Predict Base Hits in Baseball

Michael Polonio

Abstract

As the game of baseball adapts to the information age, it becomes increasingly reliant on analytical and statistical methods in order to maintain a competitive edge. Predictive models in the MLB are useful and highly sought after by teams, coaches, and the sports betting industry. In this paper I will use classification-based machine learning techniques with the aim of predicting the result of an at-bat, whether it be an out or a hit, by using information about the pitch. Classification will be done using a Support Vector Machine, Naïve Bayes Classifier, Decision Tree, Random Forest, and a Deep Neural Network. The models will then be evaluated and compared to each other. Previous researchers have had success with these machine learning methods, so my goal is to find out if they perform comparably well when applied to the problem of predicting at-bat results.

Introduction

Baseball, known as the American Pastime, has been played for about 150 years and is one of the Major professional US sports. Major League Baseball (MLB) is the premier professional baseball league in the world. Due to its discrete nature and large amount of data, baseball lends itself as a great subject of statistical analysis. Analysts can now apply machine learning algorithms to large baseball datasets to derive meaningful insights into player and team performance (Koseler & Stephan, 2017).

Major League Baseball (MLB) has been a topic of increased research and analysis in recent years. The popularity of baseball and the size of the business has made it a sought-after topic to produce predictive models in hopes of being able to predict outcomes that occur in the sport, whether it be which pitch will be thrown to the batter, the association between active spin rate and the opponents' batting average, predicting win-loss outcome of a game, or general analytics methods used to predict various outcomes that occur in the sport.

In this paper I will attempt to develop models that can predict whether an at-bat will result in a hit or an out using information about the pitch (speed, spin rate, etc.) .

Literature Review

The basis that all baseball research is based on is called Sabermetrics, or the science of learning about baseball through objective evidence and statistics (Albert, 2010). Albert discussed the different statistics in the game of baseball and how they are calculated. He also predicted that the technology that was being developed at the time to record live data during MLB games would lead to greater insights into the game of baseball.

With machine learning and deep learning tools becoming more accessible it has been an increasingly popular method used by researchers when performing studies in professional sports. These methods can be used to gain insight into different aspects of player and team performance. The general applications of these methods were studied by Koseler & Stephan (2017). They found it appropriate to classify the problems of baseball analytics as Binary Classification, Multiclass Classification, and Regression. They found the most used algorithms on large baseball datasets were Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Bayesian inference. SVMs and KNNs were used in at least 25% in the studies that met their criteria. Artificial Neural Networks we used in 9% of the studies. The results also indicated that SVMs were the most popular method for classification tasks while Bayesian inference mixed with Linear Regression was the most popular for Regression tasks, while Bayesian inference can be used for both Classification and Regression. They also predict growth in the popularity of research at the intersection of baseball analytics, machine learning, and neural networks.

Teams can benefit from the application of machine learning by using the results of the models to help them make better on-field decisions. One important aspect in the game of baseball is deciding when to replace a starting pitcher with a relief pitcher. This is the question that the research of Ganeshapillai, & Guttag (2014) aimed to address. They identified this problem as a classification problem of predicting whether a starting pitcher would give up at least one run if allowed to start the next inning but transformed the problem into a Regression problem by using Pitchers Total Bases (PTB) instead of runs allowed as the dependent variable. For predictors they used information about the current at-bat, game situation, and historical data. Their findings indicate that games in which the manager left a starting pitcher in the game that their model would have removed, the pitcher surrendered a run 60% of the time in the next inning even though runs are surrendered in only about 10% of innings.

Valero (2016) used methods such as artificial neural networks (ANN), support vector machines (SVM), decision trees, and lazy learners to perform a study on predicting win-loss outcome in MLB games. He found that all these methods performed at just under 60% accuracy, nearly a 10-point improvement from guessing.

Huang and Li (2021) compared different deep learning and machine learning models to predict the win-loss outcome of MLB games and investigated the differences between the models in terms of their performance. Different algorithms that were used include artificial neural networks (ANN), convolutional neural networks (CNN), and support vector machines with fivefold cross validation. The models performed with accuracies of 94.18% (CNN), 94.16% (ANN), 93.90% (SVM). The models in this study performed with higher accuracy than related studies, and it was also the first time a CNN was used to predict the outcome of MLB games.

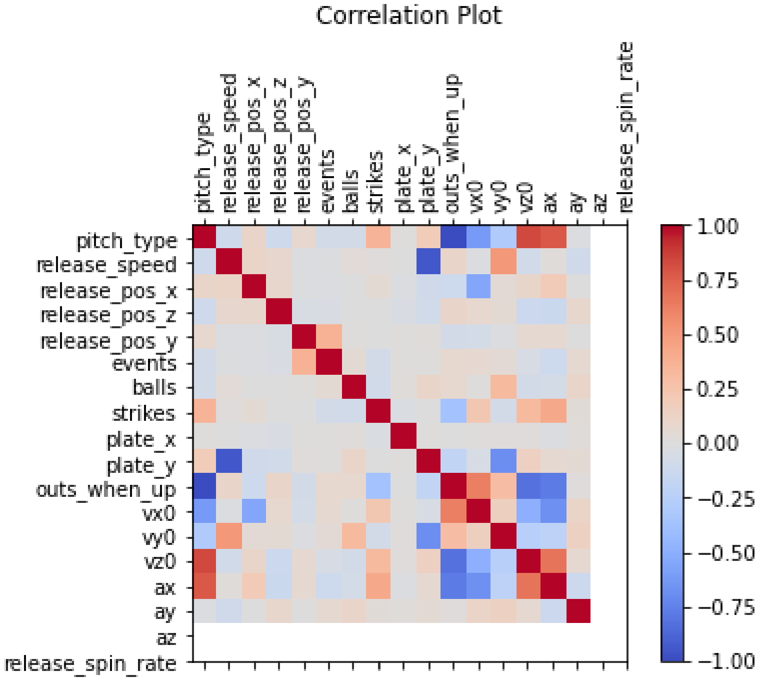
The ability to be able to predict which pitch will be thrown by the pitcher can be a valuable asset to teams and their training staff. A hitter has the advantage over the pitcher if he knows what pitch is coming. The problem of coming up with a machine learning model for this was studied by Hamilton et al. (2014). The data they used had 50 features and they used classification methods to predict pitches. Their model used post-pitch information about a pitch to classify the pitch and pre-pitch information to classify its type. A Support Vector Machine and k-Nearest Neighbors was used for classification. They found that these algorithms were able to predict pitches with a 79.76% (SVM) and 80.88% (k-NN) accuracy. Since this model represents a significant improvement over simple guessing it is a useful tool for batting coaches, batters, and anyone who wishes to more deeply understand the dynamics of a pitching matchup.

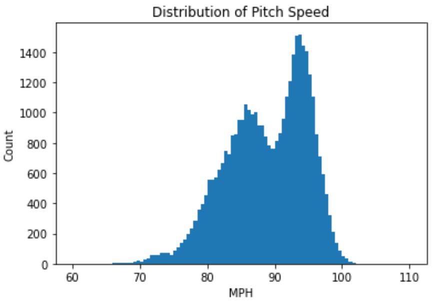
Not only is sports itself a large and lucrative business, but so is sports betting. Being able to predict the winner of a match or a championship can have huge financial implications in the world of sports betting. Tolbert and Trafalis (2016) used a Support Vector Machine as their model to predict the American League Champions, National League Champions, and World

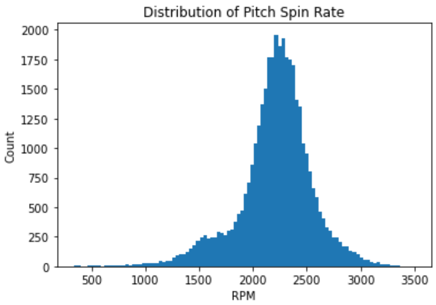
Series Champions at a higher success rate compared to traditional methods. They applied different SVM algorithms to determine the best predictor of championship winners, such as linear, quadratic, cubic, and Gaussian kernels. They were able to achieve up to 69% accuracy on their models and successfully predicted the winner and loser of the 2015 World Series.

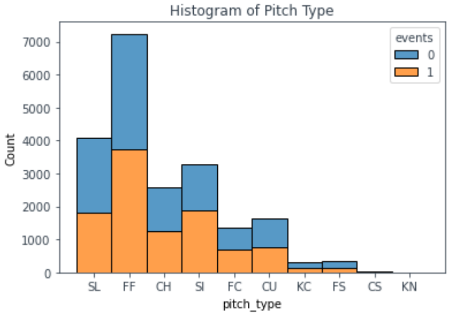
Due to the large amount of data and discrete nature, baseball is a popular topic of analysis. Teams can directly benefit from the results gathered by machine learning models. It can be useful for a team, coaching staff, or someone in sports betting to be able to predict what the result of an at-bat will be. In this paper I will utilize the algorithms used by past researchers and apply them to the problem of classifying at-bat results using the pitch data as predictors. All the different models used will be evaluated and compared.

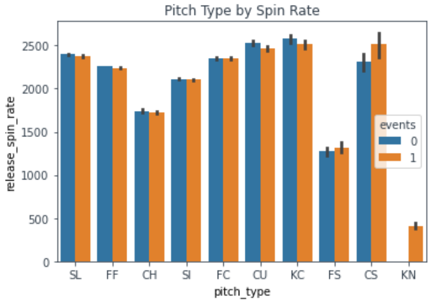
Exploratory Data Analysis



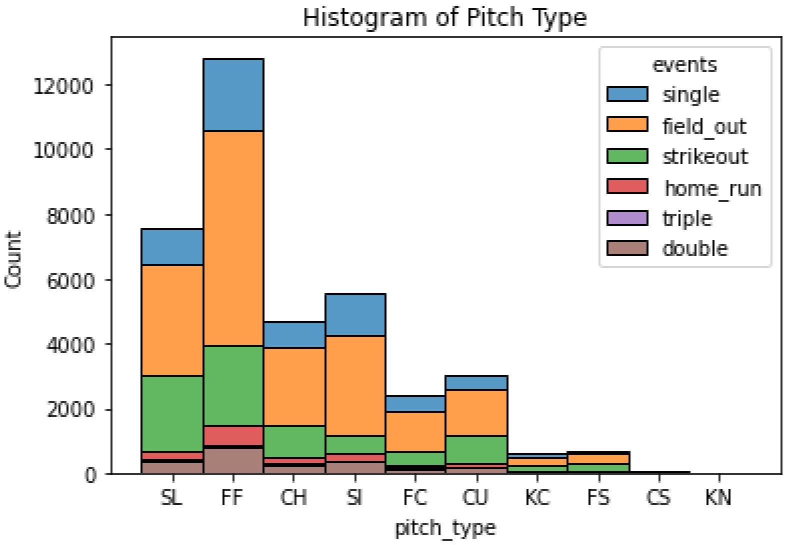

Distribution of Pitch Speed

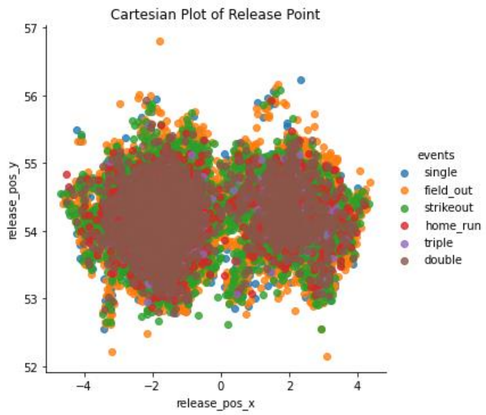

Distribution of Pitch Spin Rate

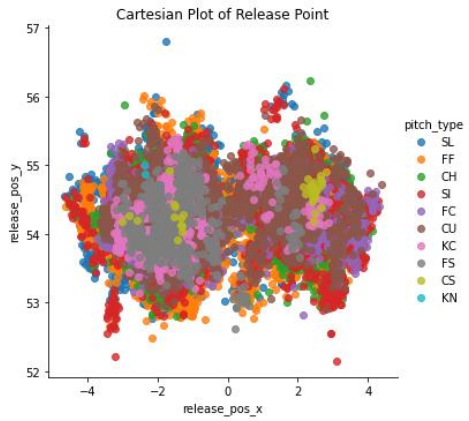

Histogram of Pitch Type

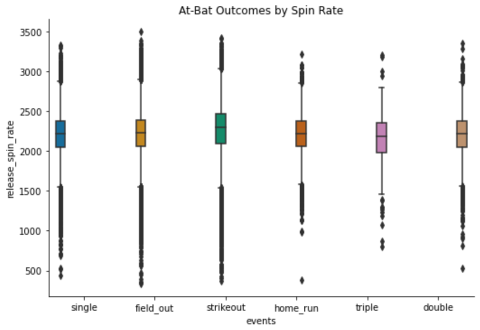

Pitch Type by Spin Rate

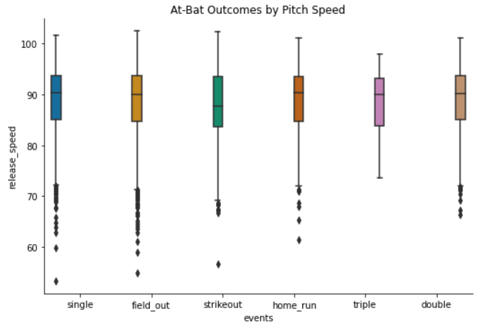
0: out 1: hit


Histogram of Pitch Type









Methods

Data Collection

The field of baseball analytics received a significant resource in 2015 when the MLB installed Statcast in every stadium. The highly accurate tool is a system of 12 cameras located around the park used to analyze many different aspects of the game and collect data on player and movements of the baseball. This technology integrates doppler radar and high-definition video to measure the speed, acceleration, and other aspects for every player on the field.

Statcast data is publicly available and are offered by several sources. The data can be queried on BaseballSavant.com where the user can specify which fields are needed.

Dataset

The dataset used to train the classifiers in this paper was collected from all MLB games in the 2021 season. Each observation represents the last pitch of an at-bat and the metadata (features) associated with it. The number of features used in this analysis was reduced to 18. Features were chosen based on their relevancy to the at-bat result. The features include pitch type, release speed (mph), number of balls and strikes, pitch spin rate (rpm), x/y/z coordinate of the release point, x/y coordinate of the where the pitch crosses the plane at the plate, x/y/z velocity and acceleration of the pitch. The pitch types include fastball, slider, sinker, changeup, etc. The coordinate of the release point pertains to the point at which the pitcher releases the ball in the plane of releasing the ball.

The target variable will be at-bat result (events). Approaching this as a binary classification problem, the target outcomes will be *hit* and *out*. Hits will include outcomes such as single, double, triple, homerun. Outs will include strikeouts and fielded outs.

The dataset used for analysis contains 37,268 observations. When downsampling was performed for an equal number of hits and outs the number of observations was reduced to 20,820.

Tools

Python will be used in this analysis. The scikit-learn and Keras libraries will be used to build the classification models. Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems (Pedregosa et al, 2011). Scikit-learn provides implementations of many of the well-known machine learning algorithms in Python, one of the most popular languages for data analysis. Python also proves itself useful for data visualization. The popular visualization library Matplotlib was used to graph the data exploration. The packages from scikit-learn that were used to build the models were the following: DecisionTreeClassifier, RandomForestClassifier, GaussianNB(), and SVC (rbf). Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models (Brownlee, 2020). Keras was used to build the neural network (Sequential).

Classification Methods

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data (Brownlee, 2020). For the purposes of this research, classification represents determining the result of an at-bat, whether the batter reaches with a hit or gets out. Classification will be done on a label of two levels: batter reaches base (1B, 2B, 3B, HR), or gets out (Strikeout, fly-out, ground-out). Feature selection will be performed to determine which features will be the most important ones to include when building the models. The classifiers will be modeled with a Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, and a Deep Neural Network. All of these models were evaluated and compared by their model accuracy, CV score, F1 score, and precision/recall.

In addition to the classical machine learning techniques mentioned above, deep learning will also be used for classification. Using Keras, a multilayer perceptron (MLP) model will be constructed.

MLP Architecture:

```
Layer (type)          Output Shape          Param #
```

```
=================================================================

dense (Dense)              (None, 64)              960

dropout (Dropout)          (None, 64)              0

dense_1 (Dense)            (None, 64)              4160

dropout_1 (Dropout)        (None, 64)              0

dense_2 (Dense)            (None, 1)               65

=================================================================

Total params: 5,185

Trainable params: 5,185

Non-trainable params: 0

_________________________________________________________________
```

The model used in this paper had an architecture that contained two hidden layers with 64 nodes each, both with ReLU activation functions and a 40% dropout to avoid overfitting. The model was compiled with an Adam optimizer with a learning rate set to 0.001, and a binary crossentropy loss function. The output layer consisted of one node for binary classification.

Results

In this paper I proposed using machine learning techniques to predict outcomes of hit/out at-bats in the MLB using information about the pitch as input variables. Figure 1 & 2 shows the comparison between the accuracy of different classification methods used in this study. As for the models that were trained on the natural sampling of the dataset (outs occur more frequently than hits): the Deep Neural Network outperformed all other classification methods with an accuracy of 72.1%. Second was the Support Vector Machine with 71.7% accuracy. The model was fit with an 0.2 validation split and set to perform 50 epochs. Even though the random forest predicted with 71% accuracy it had a very low F1 Score of 0.09, indicating that it was not good at predicting hits.

Model Name	**Accuracy**	**CV Score**	**F1-Score**	**Precision**	**Recall**
Decision Tree	62%	0.52	0.34	0.33	0.35
Random Forest	71%	0.52	0.09	0.41	0.05
Naive Bayes	69%	0.52	0.669	0.50	0.17
SVM (RBF kernel)	71.7%	0.42	0.34	0.33	0.35
Deep Neural Network	72%	0.52	0.4	0.41	0.39

Fig1

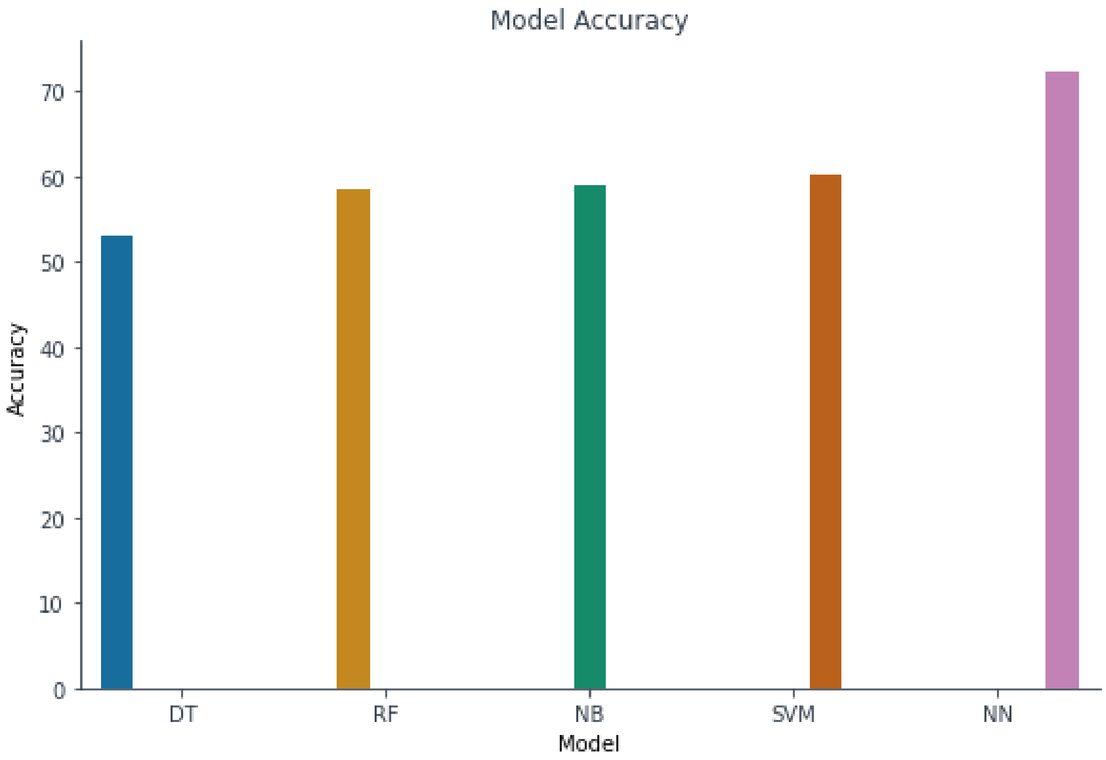


Fig 2

The Support Vector Machine may have performed to the highest level of accuracy in several other studies on the topic, but not for the problem of predicting whether the outcome of the at-bat will be a hit or an out. Although the DNN outperformed the other models, successfully predicting a hit poses to be a very difficult task. All the models performed much better when predicting outs, albeit they occur more frequently. The next part of my study addresses when the dataset has an equal number of hits and out. Down-sampling was performed on the majority class, the batter getting out. Each model was trained on a dataset with 20,820 observations, a 50/50 split for the target variable outcomes hit or out.

Results from dataset with equal amount of hits/outs:

Model Name	**Accuracy**	**CV Score**	**F1-Score**	**Precision**	**Recall**
Decision Tree	53%	0.54	0.52	0.53	0.52
Random Forest	58.5%	0.54	0.60	0.57	0.63
Naive Bayes	59%	0.54	0.61	0.57	0.65
SVM (RBF kernel)	60.3%	0.44	0.64	0.58	0.71
Deep Neural Network	62%	0.54	0.61	0.60	0.68

Fig3

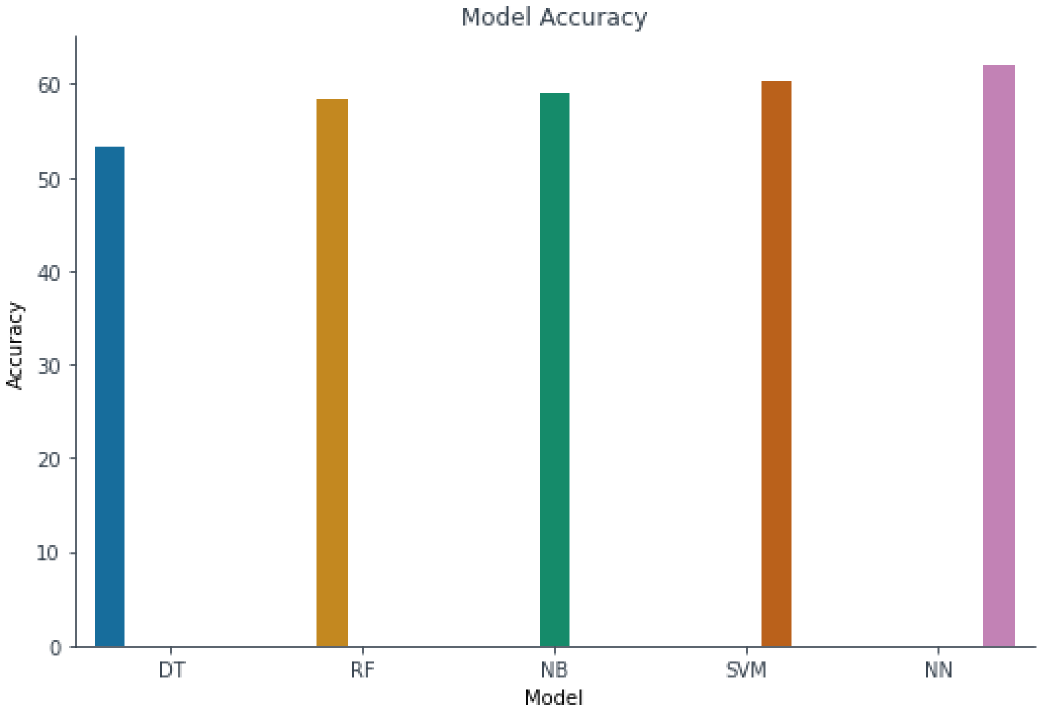


Fig 4 Model Comparison with equal number of hits and outs

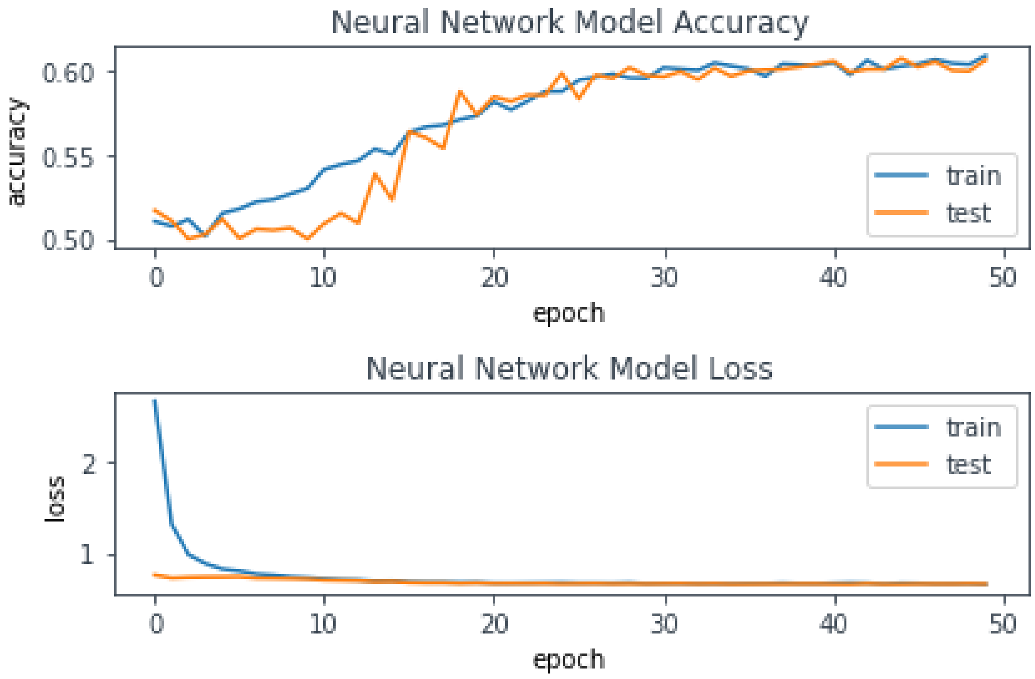


Fig 5 Deep Neural Network learning curve

The DNN outperforms all other models when looking at the balanced dataset as well. The *precision* and *recall* are improved in these models, although the accuracies are lower. This indicates that the models are better at predicting both binary levels.

The DNN improved up to 50 epochs before overfitting, with an accuracy of 62%. Although these models with an accuracy around 60% they are more reliable at predicting hits and outs than the previous models with a higher accuracy

Conclusion

Predicting whether an at-bat will result in a hit or an out based on information about the pitch proves to be a difficult task even for these machine learning techniques, but it was still possible to do so 62% of the time when analyzing the dataset with an equal amount of hits and outs. The Support Vector Machine may have performed to the highest level of accuracy in several other studies on the topic, but not for the problem of predicting whether the result of the at-bat will be a hit or an out. The Deep Neural Network (MLP) performed the highest when looking at the naturally occurring hits/outs and the equal number of hits/outs with an accuracy of 72% and 62% respectively.

I predict these models would perform better when trained on individual pitchers. Using pitch data from all MLB pitchers introduces a sort of randomness and high variability regarding pitch information. Training on individual pitchers would provide more easily identifiable differences between pitches.

I think with some improvement, these models can be useful when analyzing a pitcher to gain a competitive advantage

References

[1] Kaan Koseler & Matthew Stephan (2017) Machine Learning Applications in Baseball: A Systematic Literature Review, Applied Artificia Koseler & Matthew Stephan l Intelligence

[2] Hamilton, Michael, et al. 2014. “Applying machine learning techniques to baseball pitch prediction”. In Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods, 520–527. SCITEPRESS-Science and Technology Publica-tions, Lda.

[3] Alvarado, A., Sharp, R., Brown, A., White, E., Kusters, I. S., & Dean, J. M. (2021). MLB Statcast Pitch Analysis: The Association between Active Spin and Opponent's Batting Average. In *International Journal of Exercise Science: Conference Proceedings* (Vol. 2, No. 13, p. 50).

[4] Mahesh, B. (2020). Machine Learning Algorithms-A Review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

[5] Albert, J. (2010). Sabermetrics: The past, the present, and the future. *Mathematics and sports*, 3-14.

[6] Valero, C. S. (2016). Predicting Win-Loss outcomes in MLB regular season games–A comparative study using data mining methods. *International Journal of Computer Science in Sport*, *15*(2), 91-112.

[7] Firsick, Zachary. 2013. "Predicting Major League Baseball Playoff Outcomes Through Multiple Linear Regression". PhD thesis, University of South Dakota.

[8] Ganeshapillai, G., & Guttag, J. (2014). A data-driven method for in-game decision making in MLB. *MIT SSAC*.

[9] Attarian, A., Danis, G., Gronsbell, J., Iervolino, G., & Tran, H. (2013). A comparison of feature selection and classification algorithms in identifying baseball pitches. In *International MultiConference of Engineers and Computer Scientists* (pp. 263-268).

[10] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), 121-167.

[11] Haghighat, M., Rastegari, H., Nourafza, N., Branch, N., & Esfahan, I. (2013). A review of data mining techniques for result prediction in sports. *Advances in Computer Science: an International Journal*, *2*(5), 7-12.

[12] Huang M-L, Li Y-Z. (2021) Use of Machine Learning and Deep Learning to Predict the Outcomes of Major League Baseball Matches. *Applied Sciences*.

[13] Lage, M., Ono, J. P., Cervone, D., Chiang, J., Dietrich, C., & Silva, C. T. (2016). Statcast dashboard: Exploration of spatiotemporal baseball data. *IEEE computer graphics and applications*, *36*(5), 28-37.

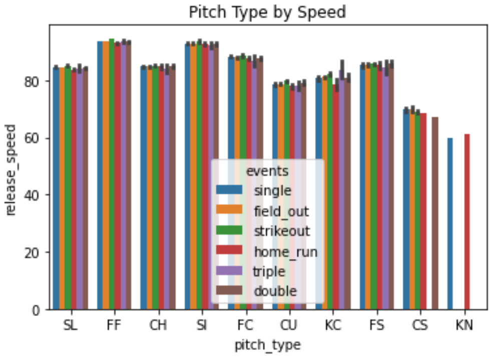
[14] Demers, S. (2015). Riding a probabilistic support vector machine to the Stanley Cup. *Journal of Quantitative Analysis in Sports*, *11*(4), 205-218.

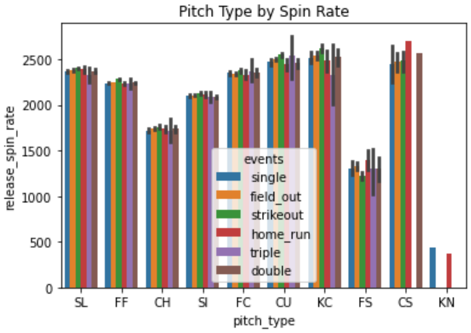
[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

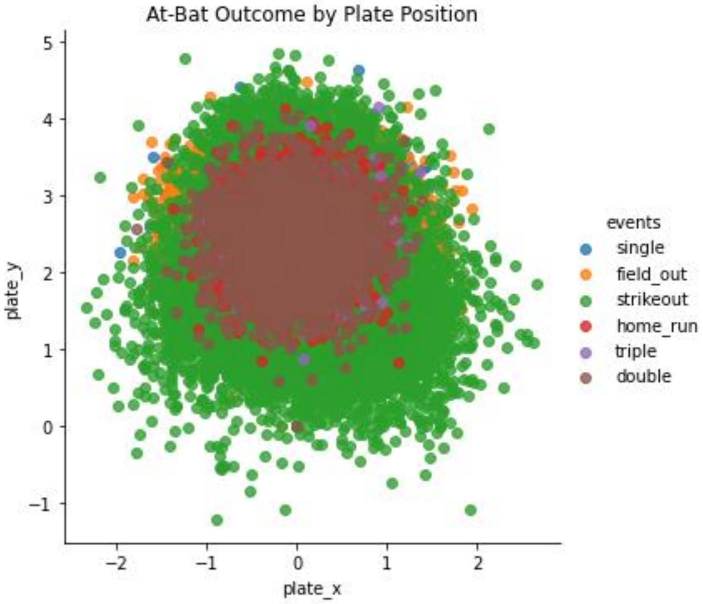
[16] Tolbert, B., & Trafalis, T. (2016). Predicting major league baseball championship winners through data mining. *Athens Journal of Sports*, *3*(4), 239-252.

[17] Brownlee, J. (2020, August 19). *4 types of classification tasks in machine learning*. Machine Learning Mastery. Retrieved November 20, 2021, from https://machinelearningmastery.com/types-of-classification-in-machine-learning/.

Appendix


Pitch Type by Speed
release_speed
0
20
40
60
80
SL
FF
CH
SI
FC
CU
KC
FS
CS
KN
pitch_type
events
single
field_out
strikeout
home_run
triple
double


Pitch Type by Spin Rate
release_spin_rate
0
500
1000
1500
2000
2500
SL
FF
CH
SI
FC
CU
KC
FS
CS
KN
pitch_type
events
single
field_out
strikeout
home_run
triple
double


At-Bat Outcome by Plate Position
plate_y
−1
0
1
2
3
4
5
plate_x
−2
−1
0
1
2
events
single
field_out
strikeout
home_run
triple
double