CISC 451 – Course Project

Feature Engineering and Supervised Learning in Professional Ice Hockey

Final Report

November 25th, 2020

Gavin McClelland – 10211444
Marshall Cunningham - 20249991

Table of Contents

List of Figures

1.0 Introduction, Background, and Problem Definition

Data analytics has been rapidly adopted in the realm of professional sports over the past decade and has created an increasing appetite for the application of data-driven methods to develop a better understanding of different concepts. This is especially the case is professional ice hockey, where advanced statistics started being collected by the National Hockey League (NHL) in the 2007-2008 season.

The accessibility of open-source NHL datasets allows for different analytical approaches to be explored and has created a community of data scientists who have collectively brought to light the impact analytics can have on the professional hockey landscape. However, there is a common understanding that there is much more parity in the NHL relative to other sports. As such, there exists a high demand to explore data collected by the NHL to develop novel insights regarding the different factors contributing to success (i.e. "winning"). On this topic, the NHL established a relationship with MGM as an official betting partner in 2018 (Rosen, 2018).

The problem to be addressed is to develop a better understanding of the statistical factors contributing to wins in the NHL. In this assignment, this was approached by using the wealth of data readily available in this space to predict whether a team will win. From there, this would allow for the components of a match to be deconstructed, at which point the individual impact of each feature could be analyzed in future work. This type of work bridges the gap professional teams face in getting the most out of their players and game tactics.

The initial approach specified in the proposal specified the sole exploration of event-driven data—specifically shot locations—but similar attempts in literature are much more sophisticated and are out of the scope of this project's timeline. More importantly, these previous approaches have been solely focused on determining the likelihood of a shot becoming a goal, instead of these micro-level events contributing to the outcome of a game. This likelihood is called "expected goals" (more simply, xGoals), and is included as part of the final dataset from MoneyPuck, among other probabilistic measures.

This report details the dataset and software tools used, along with the analytics methodology from start to finish to accompany the software written. There were three main approaches that were taken in attempt to solve this binary classification problem. The first was to develop a classifier using statistics from that game with no knowledge of the opponent. The second was to develop a classifier using a rolling average of statistics from the previous 1, 3, 5, and 10 games. The third approach was to use the rolling average statistics in the second approach and combine that with knowledge of the opponent to create a classifier that would predict the result of specific matchups before the game had even taken place. The best-observed results from these classifiers were 0.9024, 0.5515, and 0.5768 for approaches 1, 2, and 3 respectively.

2.0 Brief Dataset Description

The initial dataset specified in the proposal was used for the EDA phase as explored in the midterm submission, entitled "NHL Game Data", found on Kaggle (Ellis, 2019). This dataset contains multiple files from the NHL Real-Time Scoring System (RTSS), arranged as a relational database. These files encompass game outcomes, individual events (i.e. hits, shots,

face-offs, etc.), and many others dating back to the 2007-2008 NHL season. The files of interest are highlighted in the Entity-Relationship Diagram (ERD) provided Figure 1.

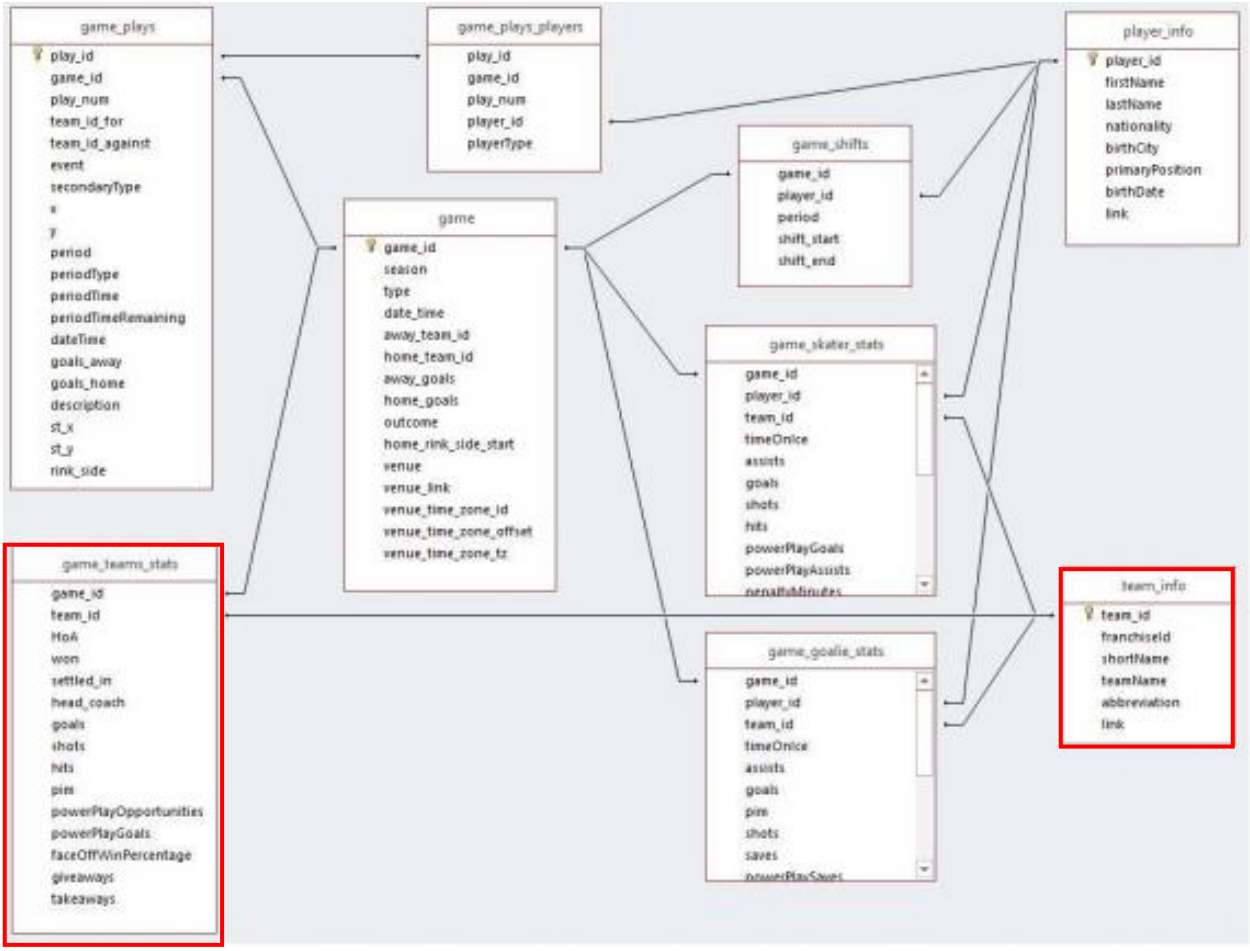

Figure 1 - ERD from provided Kaggle dataset, csvs that are of interest are highlighted in red.

This dataset was used in the majority of the EDA phase of the project, and after the submission of the midterm report, it was understood that the team-level metrics were just game-level aggregates of lower-level datasets (i.e. player-level and event-driven data). The event driven approach seemed promising at first glance, but it did not offer much in addition to the team-level dataset, and the team-level metrics were too simple to dig much deeper on. Thankfully, MoneyPuck has advanced metrics at the team-level for each gameId in the RTSS data (MoneyPuck, 2020). This was easily stitched with the game_teams_stats.csv file to resolve the outcome of each gameId. Most of the statistics from MoneyPuck including their definitions are included with the submission in MoneyPuckDataDictionaryForPlayers.csv. These features give context to each game regarding puck possession, shot share, and expected performance. This final dataset served as a basis for the preprocessing steps required for each of the three analytical approaches.

3.0 Assessment of Challenges and Obstacles

Challenges encountered throughout the project include the following:

1. Large volume of data to start with (approximately 1 GB). This was addressed by selecting an appropriate level of abstraction for analysis (team-level data instead of event-level)
2. EDA took a while because it was difficult to know what to look for. A lot of work in this space is not open-source, and some of the data sources considered were behind a paywall, such as Hockey-Reference (Hockey | Team Advanced Stats Finder, 2020).
3. Project developers used this course as an opportunity to develop a degree of comfort using Scikit-learn, pandas, and applied data science in python in general. This posed development hurdles at times, but it was a valuable learning experience.
4. Choosing a difficult problem. Knowing that this binary classification problem has been approached by developers with much more experience and the theoretical best performance is fabled to be ~62% due to parity and luck did not instill much hope from the outset of an eight-week development effort (Weissbock, 2014). However, the more important takeaway is to understand why it is such a difficult problem, which could lead to more novel insights in future work.

4.0 Methodology and Analytics Process

4.1 Software Packages and Download Instructions

All code written for this assignment was written in the Python programming language using Jupyter notebooks. The main software packages used in this assignment were, pandas, scikit-learn, and numpy. A tree structure of the submitted directory is shown below.

```
|   CISC 451 - Final Report.pdf
|   win_prediction_notebook.pdf
|   requirements.txt
+---code
|       midterm_eda.ipynb
|       win_prediction.ipynb
|       validate_model.py, plot_roc_curve.py, make_confusion_matrix.py
\---data
        data.zip
```

Instructions to replicate all work completed as submitted are listed below:

1. Install all dependencies by running the command “pip install -r requirements.txt” in the root directory of the submitted folder.
2. Extract the contents of the data.zip folder into the data directory.
3. Before running any notebook in the code folder, add your working directory at the top where indicated (i.e. %cd “<your directory here>”). Then, run all cells in the desired notebook. The first notebook, “midterm_eda.ipynb”, includes all work completed for the midterm submission. The second notebook, “win_prediction.ipynb” includes all work completed following the midterm submission, incorporating a more sophisticated data source and lessons learned.

4.2 EDA

Exploratory data analysis (EDA) was conducted in increasing levels of abstraction. In other words, the most micro-level datasets were explored, building up to the exploration of the highest-level dataset provided (teamstats_2017-2018_2018-2019.csv).

The first dataset explored was 'plays_2017-2018_2018-2019.csv', containing individual plays with an associated game_id, event description, and the location of the event on the playing surface, among other features. First, the different unique events were identified, then shots and were isolated to explore different properties. Shot events with a null shot type indicated that those shots either missed the net or were blocked by an opposing player, as shown in the code. The main takeaways from this analysis were the ability to visualize shot and goal locations, which was useful to indicate where the density of different events come from. Additionally, generic metrics such as the volume of shots taken, and goals scored by different shot types were created to identify different baseline metrics.

Next was a short analysis of player-level data contained in the file 'skaterstats_2017-2018_2018-2019.csv'. This dataset contains aggregates of individual events attributed to each player by game. This can be deemed useful for modeling moving forward since most features are numeric. Simple metrics were created at both the player and team-level to demonstrate that this dataset could be used to create insights at both the player and team-level. More importantly, this data could be joined with the previous dataset to create a better feature space.

The last dataset that was briefly explored was 'teamstats_2017-2018_2018-2019.csv', which was used to create a simple model in line with similar past approaches that have been published (Leung, 2018). Most published attempts at predicting game outcome have involved feature spaces at the team-level and have demonstrated that there is more room for improvement. This once again gives this project purpose in addressing this room for improvement. Two simple logistic regression models were created from this dataset to establish a benchmark model, and to explore the impact of individual features. This EDA was included as part of the midterm submission.

4.3 Preprocessing

Each of the three main approaches to address this binary classification problem used many of the same preprocessing steps resulting in a common dataframe, which was constructed as follows. First, the all_teams.csv (from MoneyPuck) is read in, limited to regular season games including and after the 2010 NHL season, then merged with the data from game_teams_stats.csv (from Kaggle/RTSS). This was used to get a correct “WON” label for each gameId from the RTSS. The “home_or_away” feature was then changed from a Boolean to a numerical 1 (home) or 0 (away), so was the label “WON”. For reasons unknown, there were 104 duplicate gameIds with the same result for both teams, so these were discarded. Next, there were initially 96 statistics that had “For” and “Against” counterparts, so the intuition was to combine these into a ratio to reduce the feature space while preserving information. These ratios were then normalized using the min-max method. The resulting dataframe is 21072x56, with two rows for each gameId, or 10536 total games-worth of data.

4.4 Approach #1

The first approach used the dataframe as-is, using information from each game with no knowledge of the opponent. The first step was to intuitively drop features that were purely categorical, or those directly indicative of the result (such as goalsRatio). Then, a validation set was used in conjunction with a default logistic regression classifier to resolve the most important features to select (using their coefficients). The drop-off in importance appeared to be feature 19, so any feature below this threshold value was dropped. The values of coefficients in descending order is shown below in Figure 2.

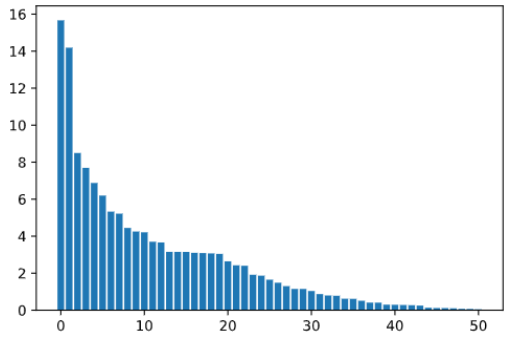

Figure 2 – Coefficient magnitudes for the feature space in approach #1.

Next, a correlation heatmap was constructed to visualize any redundant features that remained. This heatmap is shown below in Figure 3.

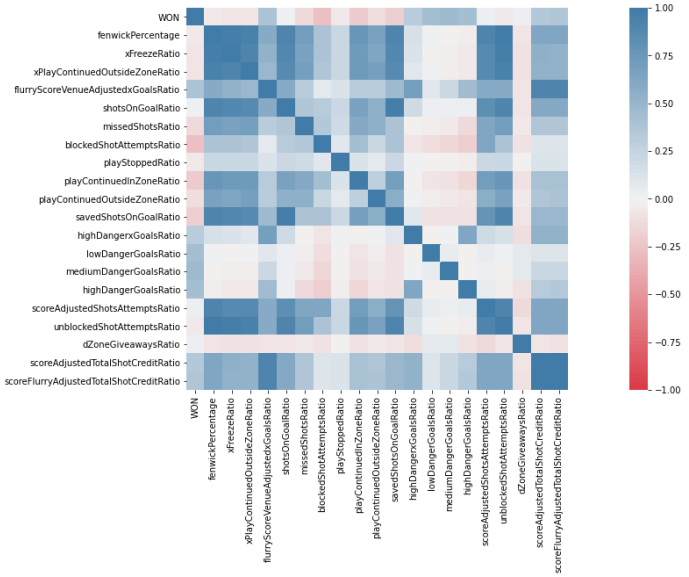

Figure 3 – Correlation heatmap for approach #1 after feature selection.

The only duplicate feature in this heatmap are the two near the bottom concerning "TotalShotCreditRatio". The decision was made to keep the score and "flurry" adjusted version, as "flurry" is regarded as a more repeatable measure with more predictive power. Next, this dataframe with the selected features was split into training and test sets, where the training data consists of all records before the 2018-2019 season, and the test set consisted of all records from the 2018-2019 season. In other words, models were trained on 8 seasons of data, and tested on 1 season. Note that the 2012-2013 season was a shortened season (48 games played per team instead of 82). A benchmark classifier was used in conjunction with 10-fold cross validation, and

the best estimator was tested against the test split. Other models were used in comparison to assess relative performance.

4.5 Approach #2

The second approach was to predict the outcome before a given game had occurred with no knowledge of the opponent. So, this required information about the team before a game were to take place. Using the dataframe from section 4.3, rolling averages of each numerical feature were created using windows from the previous 1, 3, 5, and 10 games. This of course resulted in four-times the number of features, so the same feature selection process as in approach #1 was employed, whereby the top 20 features were included before the performance drop-off. This is shown below in Figure 4.

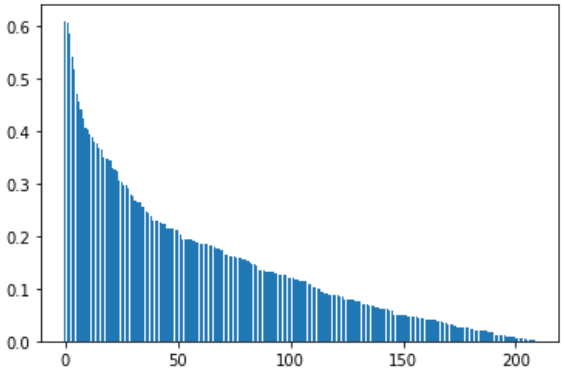

Figure 4 - Coefficient magnitudes for the feature space in approach #2.

Note the much lower coefficient values on the y-axis; this was an immediate indicator of poor predictive power. This hunch was further amplified by the correlation heatmap, where the correlation magnitudes of all features with the win column appear to be very small. This is shown below in Figure 5.

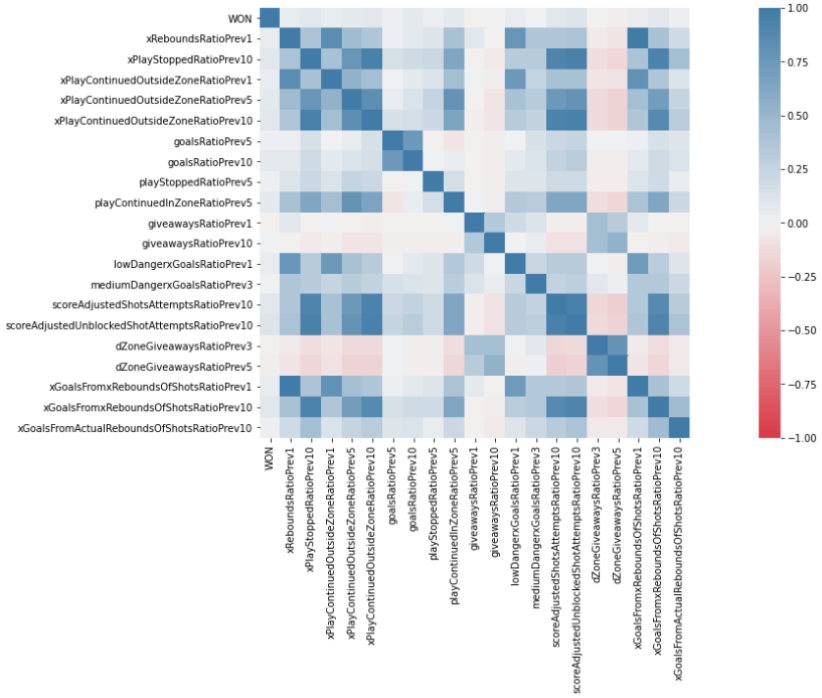

Figure 5 – Correlation heatmap for approach #1 after feature selection.

The correlation magnitude of each feature with the "WON" label is very low—and these are supposed to be the 20 best features. The same training and test splits were created as done in approach #1, and a benchmark classifier was used in conjunction with 10-fold cross validation to use the best estimator on the test set. Once again, other models were used for relative comparison. After poor accuracy was achieved, principle component analysis (PCA) was used in attempt to train a model that would generalize better to a test set, but this did not work as planned—in fact, accuracy was worse. Hyperparameters were optimized in attempt to improve performance, where a slight improvement was observed.

4.6 Approach #3

The third and most useful approach was to stitch the two records containing the same gameId together to provide a wholistic view of each record. This builds off the work done in approach #2, whereby the sliding window metrics for each team playing in each gameId are merged into one. This used the features selected in approach #2, which streamlined the process. The tricky part was merging the two records for each gameId. An XGBoost classifier was used in conjunction with 10-fold cross validation, followed by hyperparameter tuning.

4.7 Other Approaches Considered

At the outset of this term, a thought of a project to pursue was to predict the score of a game based on shot locations and compare it with the actual result. Unfortunately (or fortunately), this is exactly how expected goals (xGoals) models work. Moreover, these models are much more sophisticated than initially anticipated, and that there is much work to be done to construct a successful expected goals model before match outcome can be considered as a next step (EvolvingWild, 2018). Additionally, this is a different classification problem altogether, which was not apparent at the time of proposal. So, since this feature is included in datasets from the other sources previously mentioned (MoneyPuck), the decision was made to supplement existing work by creating a better feature space in developing a binary classifier to predict the outcome of a game, which is already known as a daunting task.

5.0 Evaluation of Work Completed

5.1 Basic Midterm Approach

Work completed for the midterm submission used simple team-level metrics to predict the outcome of a game. The first attempt using a simple logistic regression classifier is shown below in Figures 6 and 7.

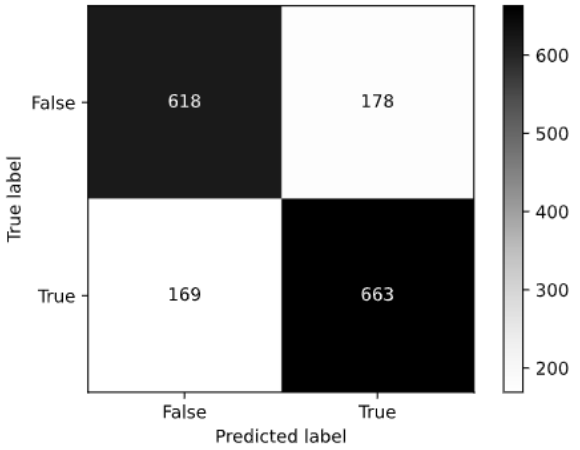

Figure 6 – Confusion matrix for the first logistic regression classifier.

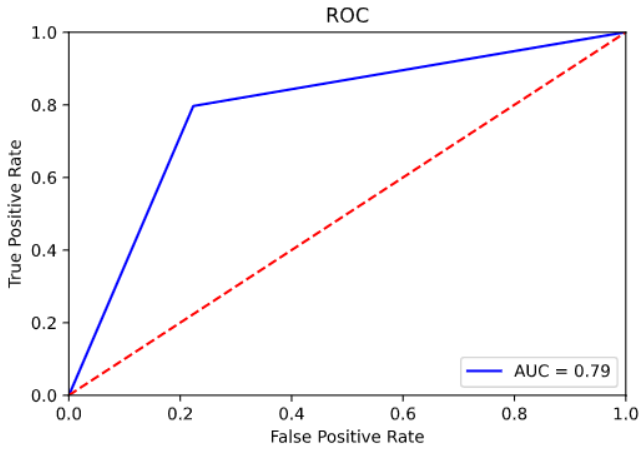

Figure 7 – ROC plot for the first classifier.

This initial ROC was high because the classifier included both goals and powerplay goals as part of the feature space, which makes sense. This essentially meant "score more goals to win more games", which is obvious. Removing these features yielded a more sensible result, shown below in Figures 8 and 9.

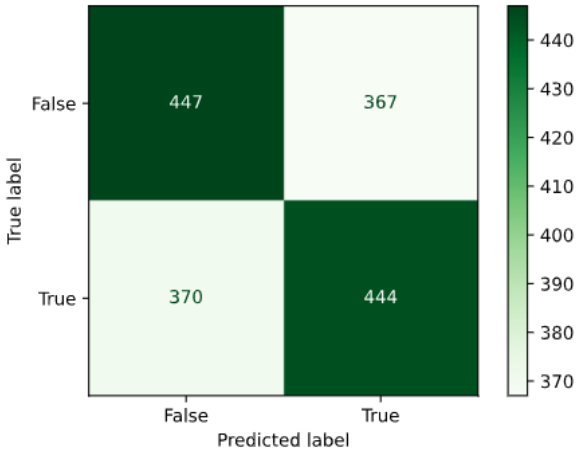

Figure 8 – Confusion matrix for the second logistic regression classifier.

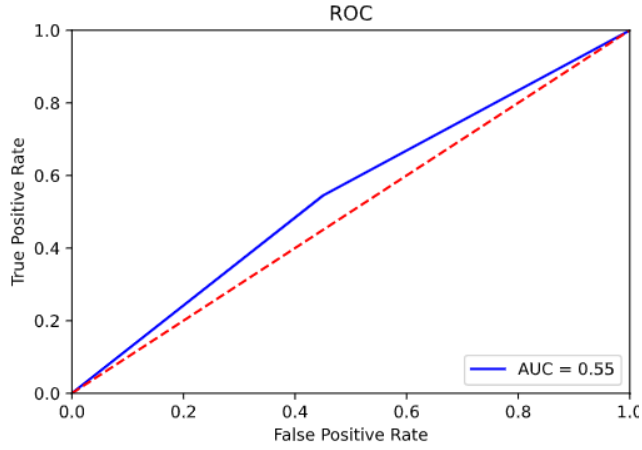

Figure 9 – ROC plot for the second classifier.

5.2 Approach #1

Using 10-fold cross-validation on the training set, the best of the 10 Random Forest estimators achieved an accuracy score of 93.31% on the test set. This same model achieved an ROC AUC of 0.9024, which is shown below in Figure 10.

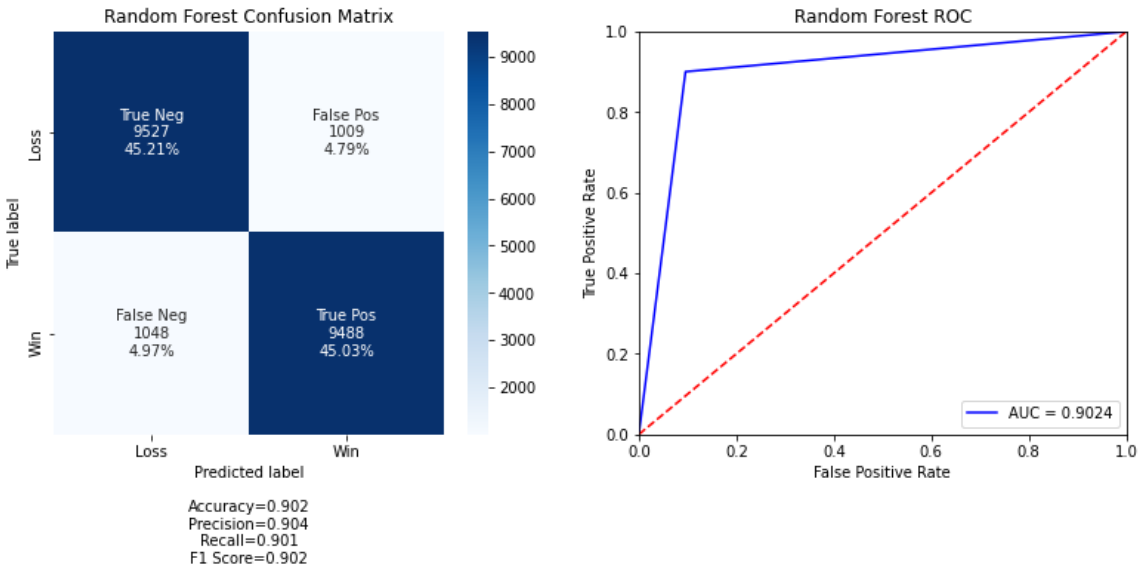

Figure 10 – Best results using approach #1, with a Random Forest Classifier.

Other classifiers were used for comparison (KNN, Decision Tree, Logistic Regression), but none of them achieved a better ROC AUC score than the Random Forest Classifier (second best was Logistic Regression at 0.8851). These results are much better than those obtained in the midterm exercise in Figure 7. Extensive parameter tuning was not performed because the bigger focus was on approaches #2 and #3.

5.3 Approach #2

Using 10-fold cross validation on the training set, the best of 10 SVM estimators achieved an accuracy of 53.76% on the training set, which is rather poor. This prompted the use of PCA in case the data were not linearly separable, but this did not work very well either (same accuracy, uncertain as to why). Then, different models were compared, among which the gradient boosting classifier performed similarly to the logistic regression model (~55% accuracy). After tweaking hyperparameters, the performance of the gradient boosting classifier improved slightly, to an accuracy of 55.2%, and an ROC AUC of 0.5515, shown below in Figure 11.

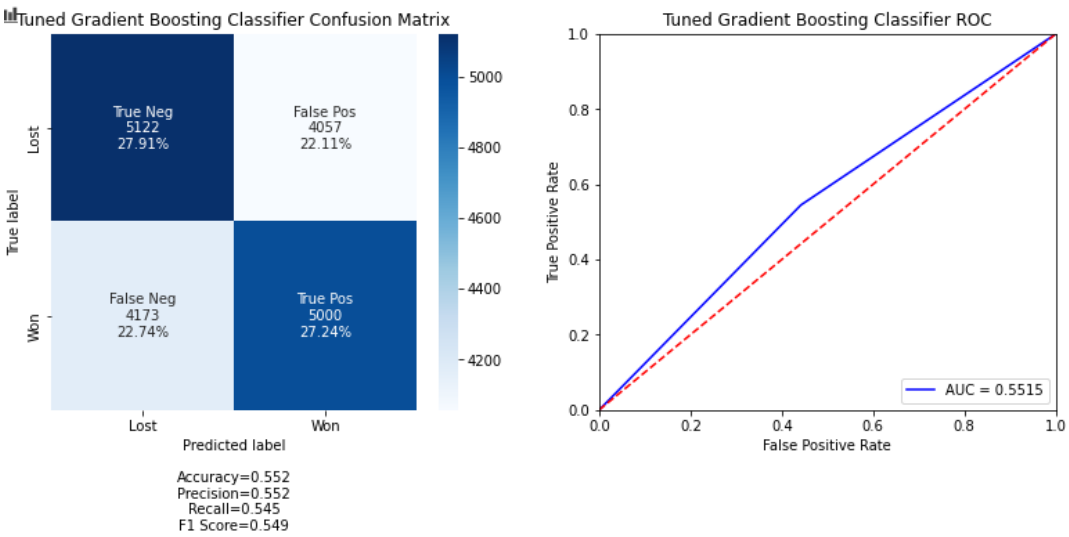

Figure 11 – Best results using approach #2 after tuning parameters on a Gradient Boosting Classifier.

Once again, other classifiers were used, but none achieved better performance than this gradient boosting classifier. This result was expected because we are no longer using information from the game itself, just the averages over the previous 1, 3, 5, and 10 games. It was surprising to observe a result that was better than chance, despite it being poor, nonetheless.

5.4 Approach #3

Training a simple logistic regression model on this dataset obtained a validation accuracy of 55.5%, and an accuracy score on the test split of 57.3%. It was strange to see an accuracy score on the test set higher than the validation score, perhaps noise from a rich feature space played a factor? The best ROC AUC was obtained with the XGBoost module, which was 0.5681 during validation, shown below in Figure 12.

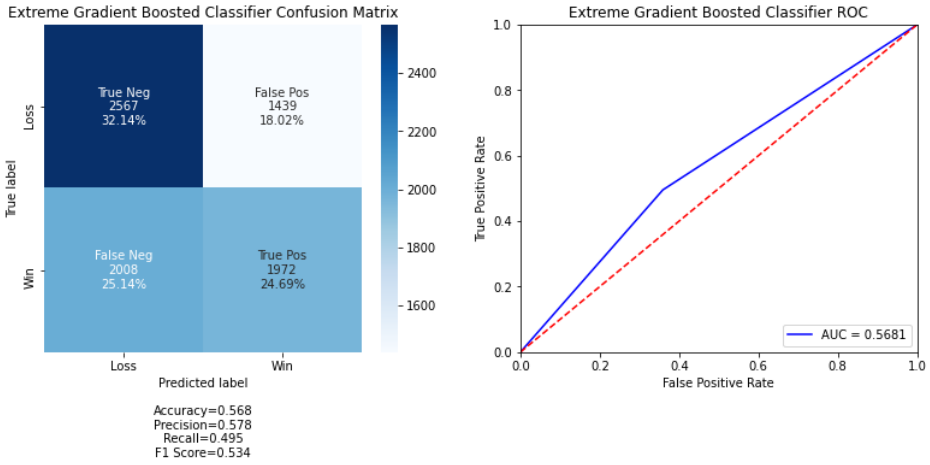

Figure 12 – Best results using approach #3 after tuning parameters on an XGBoost Classifier.

After hyperparameter tuning, the best result achieved was a score of 0.5768 ROC AUC on the test set.

6.0 Future Work and Conclusions

The clear takeaway from the work done prior to the midterm submission was that a more-sophisticated effort needed to be made to predict the outcome of a game that has yet to happen based on prior information. This proved to be a difficult task, seeing as enriching the dataset with advanced stats and features did little to improve this benchmark metric. However, introducing information about the opponent did improve this performance slightly, with an ROC AUC result of just below 0.58.

Another perspective is that predicting the outcome with information from the game can help identify the different features that are directly correlated with winning. This could serve as immediate feedback to both teams and players.

Remaining approaches that could be explored in the future are developing different versions of advanced statistics to create novel perspectives on the game. In addition, understanding the impact noise has on models of this nature. To predict using previous information, features increase exponentially. Perhaps there is a better way to investigate the past using this feature space, or perhaps this problem really does have a theoretical upper limit of 62% (where 38% is left up to "luck".

References

Ellis, M. (2019, June 21). *NHL Game Data*. Retrieved from Kaggle: https://www.kaggle.com/martinellis/nhl-game-data

EvolvingWild. (2018, June 7). *A New Expected Goal Model for Predicting Goals in the NHL.* Retrieved from rpubs: https://rpubs.com/evolvingwild/395136/

Leung, M. (2018, Aug 27). *$4,718 — Using Machine Learning to Bet on the NHL.* Retrieved from Medium: https://medium.com/coinmonks/4-718-using-machine-learning-to-bet-on-the-nhl-25d16649cd52

MoneyPuck. (2020, September 21). *Download Player and Team Data*. Retrieved from MoneyPuck: http://moneypuck.com/data.htm

Rosen, D. (2018, October 29). *NHL, MGM Resorts form sports betting partnership.* Retrieved from NHL.com: https://www.nhl.com/news/nhl-mgm-resorts-sports-betting-partnership/c-301392322

Weissbock, J. (2014). *Forecasting Success in the National Hockey League using In-Game Statistics and Textual Data.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.841.8005&rep=rep1&type=pdf

Hockey | Team Advanced Stats Finder. (2020, November). Retrieved from Hockey Reference: https://stathead.com/hockey/tpbp_finder.cgi

Appendix A

Project Plan

A detailed project plan is shown in the below Gantt chart (Figure 13). This timeline has been refined since the midterm submission as the scope of the project took shape. Following the midterm submission, more datasets were explored to supplement existing work. The process used for feature selection was completed shortly after midterm, and better features were engineered from the MoneyPuck dataset to develop models for comparison.

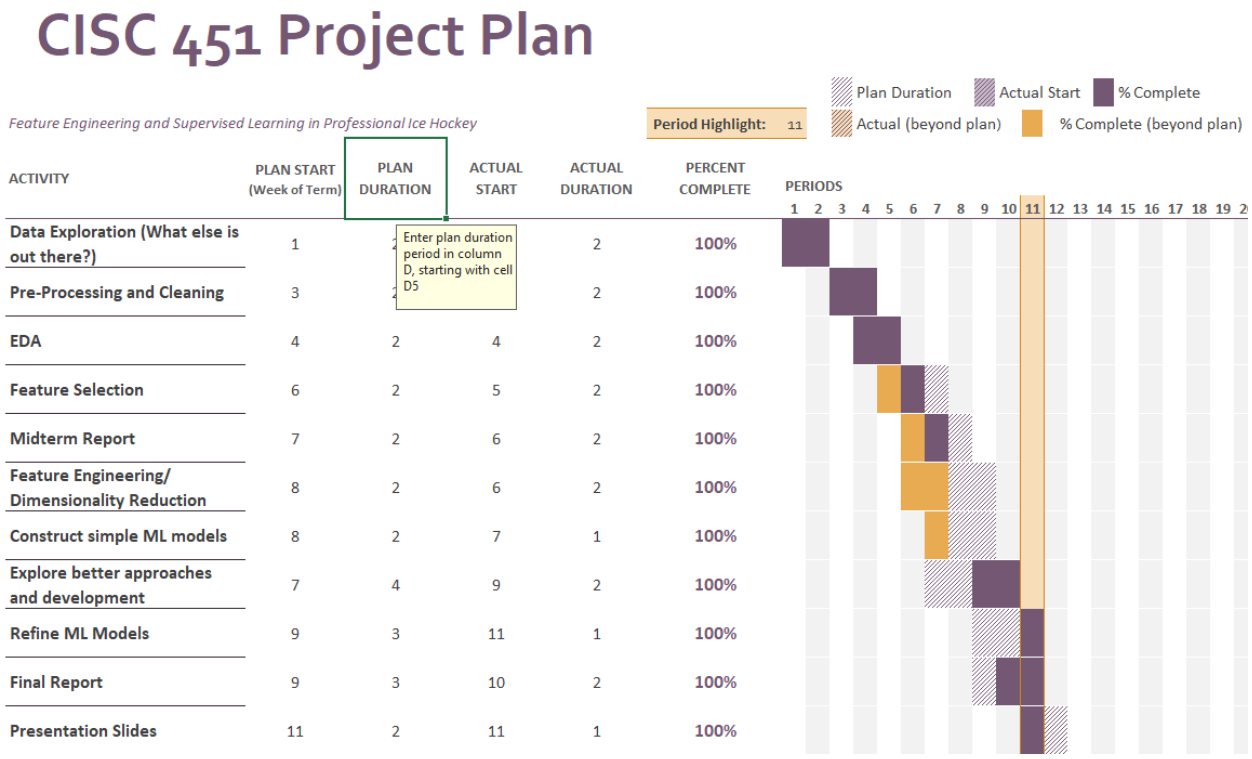

Figure 13 – Detailed project timeline at completion of the project

Description of Features

xGoalsPercentage: Expected percentage of goals scored in relation to the other team
corsiPercentage: Ratio of shot attempts including blocks
fenwickPercentage: Ratio of unblocked shot attempts
xOnGoalRatio: Ratio of expected unblocked shot attempts that are expected to be a shot on goal (not miss the net) given the context (distance, situation, etc) they were taken from. This assumes the player has average shooting talent.
xGoalsRatio: Expected goals to be scored
xReboundsRatio: Expected rebounds
xFreezeRatio: Expected number of goalie "freezes" after an unblocked shot
xPlayStoppedRatio: Expected number of play stoppages after an unblocked shot
xPlayContinuedInZoneRatio: Expected number of times the play continues in the offensive zone after the player's shot besides an immediate rebound shot. This is proxied by another event happening in the zone after the shot (such as a hit, takeaway, etc) without any events outside of

the zone happening in-between and all the same players for both teams are still on the ice as they were for the original shot
xPlayContinuedOutsideZoneRatio: see above, but outside the zone
shotsOnGoalRatio: shots on goal vs. the other team's total
missedShotsRatio: amount of shots that miss the net
blockedShotAttemptsRatio: amount of shots that are blocked
shotAttemptsRatio: amount of attempted shots
reboundsRatio: rebounds generated from shots on goal
savedShotsOnGoalRatio: saved shots on goal
savedUnblockedShotAttemptsRatio: saved unblocked shot attempts
lowDangerShotsRatio: Low danger shots (<8% xGoal value)
mediumDangerShotsRatio: Medium danger shots (Between 8% and 20% xGoal Value)
highDangerShotsRatio: High danger shots >20% xGoal value

scoreAdjusted metrics account for different leads throughout the course of a game as well as home-ice advantage. flurryAdjusted metrics discount the expected goal value of the 2nd, 3rd, 4th, etc shots in a flurry of shots. These shots are discounted because they only had the opportunity to occur because the team did not score on a previous shot

All other features are versions of each other or trivial, see attached dataset if interested.