Metrics for Goal Prediction in NHL Ice Hockey Games

Kieron Wesley

Supervisor: Dr Paolo Turrini

Third Year, 2022-2023

Department of Computer Science
University of Warwick

Abstract

The prediction of game results is commonplace in many sports, most notably with the use of the Expected Goals (xG) metric in football measuring the probability of a goal from a scoring opportunity. Ice hockey however has been slower to adopt such practices due to the more recent start of recording in-depth game data. Luck also has a greater influence on the outcome of hockey games compared to other sports, often accredited to the increased speed of play. Some metric is therefore required to quantify this to maximise the accuracy of game prediction. Using data from the 2013/14 – 2021/22 NHL regular seasons, obtained from the NHL Stats API, we attempt to identify features that can capture this notion of luck. With their inclusion in machine learning binary classifiers, we achieved an increase in predictive accuracy on past work to 69.2% using Neural Networks through iterative feature engineering. Analysis of these results and manipulation of the dataset concluded that past season performances have little influence on future games, and each season can be considered as independent. The low accuracy limit suggests that luck does have an inherent influence on game outcome that cannot be easily quantified for inclusion in classifiers.

Keywords:
Machine Learning, Neural Networks, Game Prediction, Feature Engineering, Game Simulation, Ensemble Methods, Ice Hockey

Acknowledgements

I would like to thank my supervisor Dr Paolo Turrini for his guidance and helpful insights throughout the development of this project. His suggestions proved invaluable when considering viewpoints not biased from becoming engrossed in the sport of ice hockey.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Brief Introduction to Ice Hockey

Ice hockey (often referred to simply as 'hockey') is globally one of the most popular winter sports with the National Hockey League (NHL) in North America often being considered as the pinnacle domestic league. Originating in Canada in 1917 [5], the league expanded to include teams from across both Canada and the United States by 1924.

As of the 2022/23 season, 32 teams compete in the League. The League is split into the Eastern and Western Conference (both consisting of 16 teams), each of which is further split into two divisions of 8 teams: Atlantic and Metropolitan Divisions (Eastern) and the Central and Pacific Divisions (Western) (see Figures 1 and 2). Teams are assigned into each division based on their geographical location; this is due to the large area that the League covers which results in significant travelling for the teams. To mitigate this, teams compete against each other a varying number of times per season: teams in their own division 4/5 times; teams in the other division in the same conference 3 times; remaining teams outside of their conference 2 times.

Western Conference	
Central Division	**Pacific Division**
Arizona Coyotes	Anaheim Ducks
Chicago Blackhawks	Calgary Flames
Colorado Avalanche	Edmonton Oilers
Dallas Stars	Los Angeles Kings
Minnesota Wild	San Jose Sharks
Nashville Predators	Seattle Kraken
St. Louis Blues	Vancouver Canucks
Winnipeg Jets	Vegas Golden Knights

Eastern Conference	
Atlantic Division	**Metropolitan Division**
Boston Bruins	Carolina Hurricanes
Buffalo Sabres	Columbus Blue Jackets
Detroit Red Wings	New Jersey Devils
Florida Panthers	New York Islanders
Montréal Canadiens	New York Rangers
Ottawa Senators	Philadelphia Flyers
Tampa Bay Lightning	Pittsburgh Penguins
Toronto Maple Leafs	Washington Capitals

Figure 1: NHL 2022/23 Season Teams

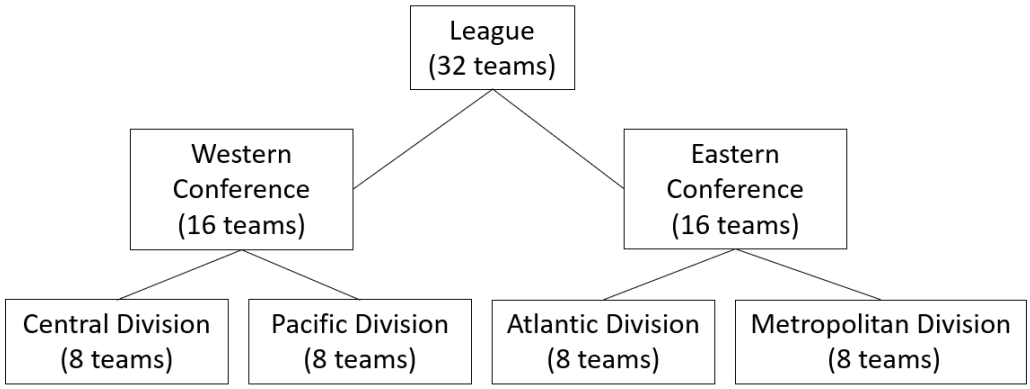

Figure 2: NHL 2022/23 League Structure

Occasionally, divisions are realigned due to the addition of new teams to the League. The most recent instance of this was at the beginning of the 2013/14 season [6] when the aforementioned divisions were created to improve the balance of competition. A temporary re-alignment was implemented for the 2021/22 season [7] due to the impact of the COVID-19 pandemic and travel restrictions imposed by the Canadian Government. The impact of this is further discussed in the *Investigating the Impact of the COVID-19 Pandemic on Game Prediction* section.

Two teams have been added to the League in the last 10 years with the Vegas Golden Knights in 2017 [8] and Seattle Kraken in 2021 [9]. 2014 also saw a name change of the Phoenix Coyotes to the Arizona Coyotes [10].

The game is played on a hockey ice rink (see Figure 3). In the NHL, the rink measures 61m/200ft long and 26m/85ft wide with boards surrounding the rink preventing the puck from leaving play unless lifted high into the air. Goals are placed 3.35m/11ft in front of the end boards, allowing play behind the net.

There are 9 faceoff dots placed around the rink where the game can be restarted from. This method of a 'faceoff' involves one skater from each team facing each other as the referee/linesman drops the puck between them.

The rink is mainly split into three zones, separated by two blue lines: Offensive zone, Neutral zone, Defensive zone. The amount of time each team spends in their respective zone is a good indication of team dominance: more time in the offensive zone suggests the team is more dominant with the puck and stronger offensively.

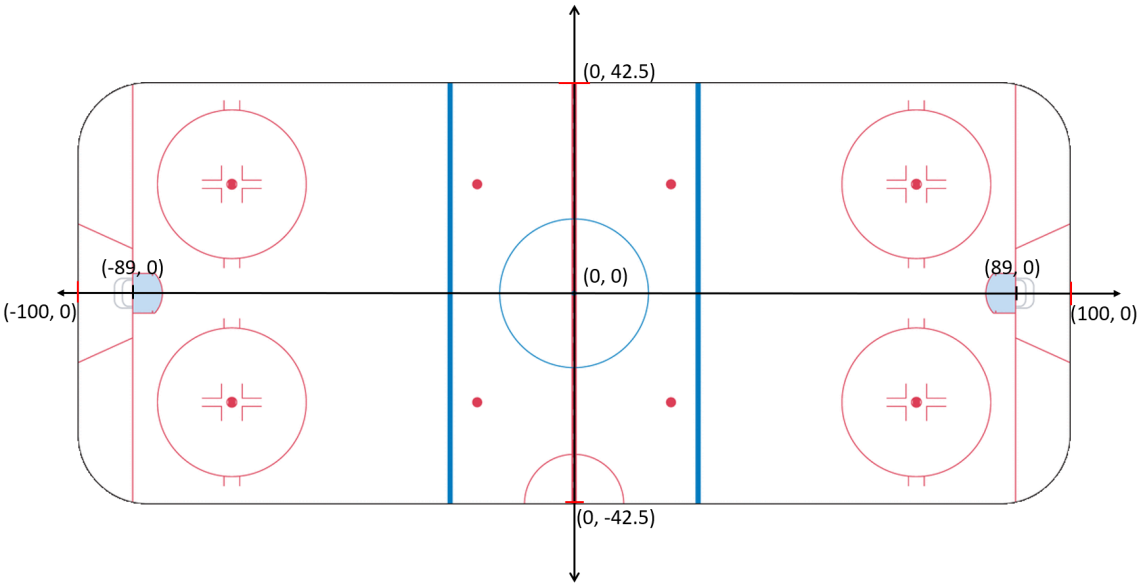

Figure 3: Hockey Rink Dimensions

The aim of the game is to shoot the puck into the opposition's goal - each goal is worth 1 point to the team's score. At any time, each team can have 5 skaters on the ice and one goalie (the goalie however can be removed for an extra skater if desired - this is often a strategy used towards the end of the game by the losing side).

Teams utilise 'rolling substitutions' with players constantly being substituted on and off the ice. This is due to the high tempo of the game requiring skaters to play at peak performance and speed during their shift. To coordinate this, a team is often split into three/four lines of 5 players. Around every minute or so, skaters will substitute for the skater in their respective

position in the subsequent line.

There are five possible positions a skater can play in: Left Wing (LW), Centre (C), Right Wing (RW), Left Defence (LD) or Right Defence (RD). These positions are often grouped into 'Forwards' and 'Defence'. Figure 4 shows the positioning of these players on the team's defensive side of the rink.

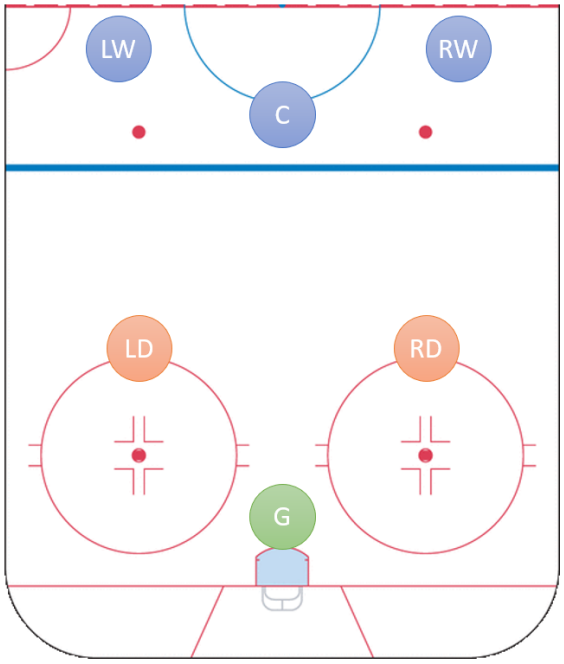

Figure 4: Hockey Player Positions

Penalties are also assessed differently for 'fouls' than in other common sports such as football. These are split into four main categories: Minor, Major, Misconduct or Match. If a player is assigned a penalty, then they must sit in the penalty box for a certain length of time during which their team looses a player on the ice (for minor/major penalties only). A minor penalty serves 2 minutes, major 5 minutes, misconduct 10 minutes and match penalty is permanent ejection from the game.

Unlike in many other sports, every game must have a winner with a draw not being possible. A game consists of three 20 minute periods with around an 18 minute interval in between. Whenever there is a stoppage in play, the clock is also stopped meaning that the total time the puck is in play is always 60 minutes. If after the end of the third period the score is tied, the game enters a 5 minute overtime. From the 2015/16 season onward, each team can ice 3 players with the first side to score winning. If however the game is still tied after overtime, a three-round penalty shootout is contested, entering sudden death until a winner is determined. The winning team earns 2 points with the losing team gaining 1 point if they lost in overtime/shootout.

The NHL regular season is played between October-April. As of the 2022/23 season, this consists of a total of 1312 games. Unlike in many other sport leagues, there is no promotion/relegation in the NHL with the regular season being played to determine who qualifies for the Stanley Cup Playoffs: the top 3 teams of each Division followed by 4 wildcards (the top 2 teams in each Conference with the most points that did not automatically qualify).

The Stanley Cup Playoffs are played immediately after the conclusion of the regular season. This is a four-round best-of-seven series with the overall winner receiving the Stanley Cup. Overtime works slightly differently during this series with full 20-minute overtime periods being played until the first team scores.

1.2 Problem Motivation

The aim of the project is to improve the predictive accuracy of Machine Learning models that attempt to solve the binary classification problem of whether a team will win or lose a given NHL regular season game.

Models that predict the outcome of sports games require large training datasets to avoid the likelihood of overfitting. Small datasets may produce

high training accuracy, by finding the optimal combination of feature weights, however fail to be generalised to new examples used in test sets. The overall accuracy of the model is therefore decreased.

The past issue with hockey was a lack of access to in-depth game data that could be used to calculate a wide range of statistics. This lack of data is attributed to the speed of play of the game. Due to a lack of appropriate technology in the early 2000s, it is assumed that people would have to have been employed to record game data post-game rather than while the game is in progress like most sports. This increased logistical effort meant that data was often restricted to simple statistics such as number of goals scored, penalty minutes conceded, shots on goal etc.

With the increase in the availability of technology, this recording process could be sped up to allow in-game data collection. This also allowed for the collection of more detailed data such as locations of individual plays on the ice, time on ice of each player, types of shot etc. As such, we now have access to much larger datasets than those available to developers of past models, therefore potentially improving the generalisation of new models (and hence, their accuracy).

It is often believed that luck has a greater influence in the outcome of hockey games compared to other sports. This would therefore place an upper limit on the predictive abilities of models since this extraneous concept is hard to quantify.

Luck is often attributed to the increased speed of play on the low-friction ice surface. This allows for pucks to be travelling upwards of 100mph [11] towards the goal. Any imperfection in the ice or minimal deflection off a player/equipment would result in an unplanned change of the puck's trajectory. Due to the speed of travel, the goalie will unlikely have enough time to react to this unexpected change, therefore increasing the chance of scoring.

We attempt to measure this luck through the statistics collected as fea-

tures in order to improve on past model accuracy detailed in the *Literature Review*. If, after this attempt, there is still a lower predictive accuracy compared to other sports, then it can be concluded that luck does have a powerful and unpredictable influence in the outcome of hockey games that mathematical models will struggle to capture.

Another feature that may limit the predictive accuracy of hockey games is the significant rate of player transfers between seasons. This often results in large changes to a squad's roster on a season-by-season basis. As such, it may be possible that team performance also varies between seasons due to changes in playing style and overall skill level. Past game data for a team from previous seasons may therefore provide less information on the probability of future game outcomes than expected. We therefore attempt to investigate this and conclude whether NHL regular seasons can be considered as independent, limiting the use of multiseason datasets.

If the project is successful (i.e. predictive accuracy is increased), then this has many potential real world applications. For example, coaching staff may benefit from knowing whether a team's current performance would predict a win/loss for their upcoming game. This could be used to determine the intensity of training required and help to focus on certain areas of their game that are lacking in quality.

Good predictions also mean that it may be possible to simulate a whole season's worth of games. This could be useful for team owners to determine how much money should be invested into the club to develop the team further and become more competitive.

2 Literature Review

A number of papers related to the field of applying Machine Learning to the prediction of match outcomes in sport were studied; not only those of past attempts at hockey game prediction but also those in other sports, most notably football.

The following four papers were the most influential in deciding the scope of this project and highlighted areas to focus on in order to further work in this area.

2.1 Use of Performance Metrics to Forecast Success in the National Hockey League (Weissbock et al.) [1]

The work by Weissbock et al. [1] from the University of Ottawa in 2013 is one of the first attempts at applying Machine Learning methods to predicting the outcome of NHL hockey games. By identifying 12 features (Table 1, also documented in the Progress Report [12]), they were able to train four classifiers that output *win/loss* for individual teams: Neural Network, Naïve Bayes, Support Vector Machine (SVM), Decision Tree.

These models were implemented with the use of the open source WEKA software [13], a collection of machine learning algorithms for use in data mining problems. This software provides an implementation of the Sequential Minimal Optimization algorithm (SMO) [14] for SVM, and the J48 algorithm [15] for Decision Trees.

Stat	**Explanation**
Goals For	Number of goals the team scored in the season so far.
Goals Against	Number of goals the team conceded in the season so far.
Goal Differential	Goals For minus Goals Against.
5/5 Goals For/Against	Ratio of goals scored and conceded when both teams have 5 players on the ice (even strength).
Power Play Success Rate	Ratio of times team scored while having a player extra on the ice (opposing team serving penalty).
Penalty Kill Success Rate	Ratio of times team scored while having a player less on the ice (team serving penalty).
Shot Percentage	Ratio of goals scored to total shots taken.
Save Percentage	Ratio of goals conceded to total shots faced.
Winning Streak	Number of consecutive games won before the current game.
Conference Standing	The team's current position in the standings.
Fenwick Close %	Ratio of the time a team is in possession of the puck compared to the opposition
PDO	Considered a measure of luck where luck implies the results of gameplays are above the league average and variance. Calculated by a team's season Shot Percentage + Save Percentage (therefore regressing to 100% average).

Table 1: Collected Features per Game by Weissbock et al. [1]

Data was collected for 517 games of the 2012/13 NHL regular season (a period of 3 months) with the use of a script that ran after every live game to add to the dataset. From this data, statistics which formed the features were calculated using a running cumulative of every game the team has played, up until the start of the recorded game.

For each of these games, 2 feature vectors were constructed: one for the home team, another for the away team. The output of *win/loss* was then determined for each team separately. The accuracy score of these models was calculated by the average 10-fold cross validation accuracy which is shown in Table 2.

Model	**10-Fold Accuracy**
NN	59.38%
NB	56.77%
SVM	58.61%
DT	55.51%

Table 2: Accuracy of Models Produced by Weissbock et al. [1]

As can be seen, the maximum accuracy achieved was 59.38% with the use of Neural Networks. This relatively low accuracy - when considering that random choice should result in around 50% accuracy - may be due to the small size of the dataset. The 2012/13 season consisted of 1230 games of which only 42% were considered by this work. As such, there may not have been enough data to accurately classify the games without the impact of overfitting.

The lack of data also meant that the only method of analysing accuracy was through 10-fold cross validation. This method involves splitting the dataset into 10 independent sets (in this example, each being on average 51 games each) then reserving one set to be regarded as the test set. The models are trained on the remaining 9 sets of data and then used to predict the output of the test set. These outputs are compared to the ground truths

of the test set to calculate an accuracy score. The process is repeated independently a further 9 times, iterating through reserving each set as the test set.

This restricts the evaluation of the performance of the models for predicting future games. A more useful approach would be to increase the amount of collected data in order to reserve a set number of games as the test set from the outset. As such, multiple models could be evaluated against the same test set to easily compare accuracies in prediction.

Another issue is the choice of feature vector implementation. By having 2 feature vectors per game (one per team competing), there is the possibility of both teams being predicted the same outcome. Although this is less of an issue in other popular sports such as football where this could be considered as a draw, it is not possible to draw a game in hockey - with the winner being decided through overtime or a shootout. As such, a winner must somehow be decided based on the strength of the confidence in this prediction; this begins to move further away from solely a classification problem and more towards a regression problem.

To prevent this issue, it may be possible to combine the two feature vectors into one vector per game. From this, only the outcome of the home team is output - this implicitly also predicts the outcome of the away team (i.e. the opposite of *win/lose*).

2.2 Assessing The Performance of Premier League Goalscorers (Green) [2]

Research into the use of machine learning to predict match outcomes began much earlier in other sports such as football due to the earlier collection of in depth statistics in the 2000's. This mainly occurred due to the larger global popularity of the sport meaning that the increased money on offer from professional success and betting companies resulted in the data analysis of the

game being far more desirable than in hockey. Also, the generally slower pace of the game meant that live data recording was easier with the technology of the time, as attendees could physically track and store statistics in-match.

From the large volume of data already available, came the first implementation of the 'Expected Goals (xG)' metric in football with the work of Sam Green [2] in 2012. This metric represents the probability of a shot resulting in a goal (in the range [0; 1]) with 0 indicating a goal is impossible and 1 that a goal is certain.

This metric was developed with the use of OptaPro data from the 2011/12 Premier League season. The data consisted of every shot attempt, with details such as shot location, shooter etc. Whilst it may initially seem intuitive that the winning team will be the side with the greatest number of shot attempts, this may not be the case if the nature of the shot quality is consistently poor. The xG metric attempts to capture this notion of shot quality by mainly analysing the shot location.

A variety of factors influence goal outcome, the two most notable being shot distance and shot angle from the goal.

The likelihood of scoring is inversely proportional to distance: the further the ball travels, the more time a defender or goalie has to block the shot and the slower the ball will travel nearer the goal due to deceleration.

The likelihood of scoring is proportional to the angle of the shot to the goal (where 90° represents a shot taken from the penalty spot and 0° from the goal-end of the pitch). A shot taken from a larger angle provides a larger target for the shooter since the goalie covers less of the goal-mouth available.

Over a course of a game, the xG of each team can be summed to determine the overall number of goals likely to be scored; the team with the higher xG should be predicted to win. This use of a regression approach for

direct goal prediction in order to classify the outcome of each team provides an additional layer to the problem as it also highlights team dominance: a larger goal difference suggests a greater skill gap.

In theory, this method could also be applied to ice hockey. However, there are a number of significant changes in terms of shooting location and scoring chance.

An increase in shooting distance in hockey does not necessarily reduce the chance of scoring, as is the case in football. There are many different types of shot available to a hockey player from the most common wrist shot to a slap shot which has greater power but reduced accuracy. This idea of different shot types influencing shot outcome is further discussed in the *Data Analysis* section.

The increase in distance also increases the chance of deflections from players in front of the net. Pucks can travel at speeds of 103mph [11]. Any slight deflection can therefore change the direction of the puck just enough to prevent the goalie from being able to react in time.

2.3 Football Analytics: A Novel Approach To Estimate Success (Kartik Narendra Jain) [3]

To further extend the traditional xG metric for football, Kartik Narendra Jain [3] from the University of Warwick increased the number of features considered in the model. This was achieved with a method of cumulatively adding feature sets such as shot-only data, player information, assist information and defensive statistics, then evaluating each incremental model.

The data was gathered from every in-game event recorded from the 2014/15 - 2021/22 Premier League seasons up until 12/12/2021. As a result, the dataset included 32 teams, 1524 players, 2818 matches and 71,170 shots.

From this, three techniques were applied: Logistic Regression, XGBoost

and CatBoost. Through 5-fold cross validation, it was concluded that the addition of features to the feature vector continuously improved model accuracy; the addition of assist information resulting in the largest increase of +3.18% accuracy, whereas defensive statistics provided the least increase of +0.11%.

This work highlights the potential improvements of predictive performance with the addition of relevant features to the feature vector, as the increase in dimensionality may provide insight into patterns that lead to winning outcomes. In addition, the large size of the dataset provides numerous examples to train the models on. This should result in increased generalisation which in turn reduces the chance of overfitting and poor test performance.

2.4 A game-predicting Expert System using Big Data and Machine Learning (Gu et al.) [4]

The more recent work of Gu et al. [4] in 2019 used an expert system approach in extension to a solely machine learning based method. A larger dataset was also utilised with records from all regular season and playoff games being collected between the 2007/08 - 2016/17 seasons. From this, 1230 games were randomly selected, equivalent to a season's worth of games at the time.

A variety of classifiers were implemented with K-Nearest Neighbours, SVM, Naïve Bayes and Decision Trees considered. In addition was the inclusion of ensemble techniques - which saw a rise in popularity during the 2010's - such as AdaBoost, which reported an increase in accuracy compared to the previous traditional methods.

Again, two feature vectors per game were constructed. This time the models were trained using the home team vectors then tested on the respective away team vectors. 8 features were identified with many statistics similar to those used by Weissbock et al. [1], however with the inclusion of

further slight variations of statistics that capture similar ideas such as puck possession.

The models achieved an accuracy of 91.8%, a significant increase in the accuracy achieved by Weissbock et al. [1] of 59.38%. This high accuracy could however be due to some arguable flaws in the team's approach. It could be proposed that the training set and testing set are not truly independent since the testing set is sampled from the same games as the training set, just using the opposing team's feature vector instead. A model that predicts e.g. *'win'* for a training example should already be predicting *'loss'* in the respective test example (vice versa), thus resulting in what appears to be a high accuracy. Instead, the test set should have been constructed by again randomly selecting games from the original dataset.

2.5 Summary of Research

The four papers discussed in this section provided the guidance necessary for approaching this project and identifying its scope.

Weissbock et al. [1] provides a good basis to start feature engineering from and can be considered a baseline for comparing model accuracy to. Any improvement on the accuracy of 59.38% will be viewed as an improvement, with 60% a good milestone to target.

The implementation of the xG metric by Sam Green [2] and the importance of shot location suggests a theoretically possible implementation on the hockey rink. This however also highlights the need to analyse the data relating to shot type and conversion rate due to the differences in game speed.

The extension of the xG metric by Kartik Narendra Jain [3] shows the potential increase in predictive power through feature engineering alone and is

therefore a main consideration in this project to help improve on previous work. More features should be considered that cover a larger scope of statistics that may not only focus solely on a team's offence - defence is also a large consideration for game outcome.

Gu et al. [4] shows how an increase in the size of the dataset may be of benefit when training the models. This also highlights the importance of ensuring the test set is truly independent of the training set, otherwise the model accuracy may be skewed to appear more powerful than in reality when applied to new examples.

3 Project Management

3.1 Tools

This section discusses the various tools used during the development of the project and reasons for their selection.

Python

The scripts used to collect game data, conduct data analysis and implement the models were written in the Python programming language [16]. This was chosen due to the numerous software libraries available to improve data processing since the success of the project relies heavily on the manipulation of NHL game data.

Three main libraries utilised were: Pandas [17] to temporarily store collected data in Dataframes (similar structures to tables in a database); matplotlib [18] to visually analyse data through various graphs; scikit-learn [19] which provides implementations of various machine learning algorithms to help construct models.

NHL Stats API

Data from every NHL game and information on individual players is accessible through the NHL Stats API [20]. This data can be accessed through a web browser, with data being returned in the JSON format [21]. Different information can be selected by manipulating the URL - more on this is discussed in the *Data Collection* section.

Microsoft Excel

CSV files were used to store the collected data, each file relating to a different category/topic of data. These could be accessed through Microsoft Excel [22] for quick inspection to ensure that data collection scripts were executing as designed and for fast visual analysis of data to highlight general points to be

explored further.

This also served as a local copy of all the data on a personal device, reducing the risk of data loss.

Kaggle

The collected data was also stored as a new dataset on Kaggle [23]. This provided an online copy of the data, improving data redundancy and reducing the risk of data loss. Kaggle also provides support for accessing and manipulating data from the dataset through Kaggle notebooks [24]. For this reason, the data analysis and model implementation scripts were written using this service.

Scikit-Learn

The scikit-learn library [19] provides implementations of many common machine learning techniques and algorithms, such as Decision Trees and Neural Networks, which allows for the fast creation of new models. Hyperparameter tuning of these models is also supported by the library to maximise model accuracy. The use of this library is discussed in the *Machine Learning Models* section.

GitHub

All data collected and scripts created were regularly backed up with commits to a GitHub [25] repository. This mitigated the risk of data loss by providing an additional storage location and also meant that data could be accessed from multiple devices. This was essential due to the failure and replacement of a personal device during the project development.

DCS Batch Compute System

The University of Warwick's Department of Computer Science provides a Batch Compute System [26] which allows for the remote running of scripts

on high performance machines. Due to the size of the training sets, this was essential to successfully train the models in an appropriate amount of time.

This would not have been possible on a personal device, with models such as Gradient Boost requiring over 27 hours to train using five-fold cross validation grid search.

3.2 Project Timeline

The project was split into three main sections:

- Data Collection and Analysis
- Model Implementation and Evaluation
- Possible Extensions

An iterative approach was utilised with progression onto the next section being determined by the success of the previous.

Focus was placed on the first two sections with the extensions being implemented depending on available time and success of the models. The original timeline proposed at the start of the project in the Specification [27] is shown in Figure 5. This was produced using the online scheduling tool monday [28] with the idea that each model will be implemented and evaluated for success before moving onto the next.

Tasks

Specification Writeup	Oct 3 - 12
Research	Oct 3 - 23
Data Collection	Oct 24 - 27
Data Analysis	Oct 28 - Nov 6
Progress Report Writeup	Nov 7 - 16
Game Winner Classifier	Nov 17 - Dec 10
Evaluate Model	Dec 10 - 11
League Simulation	Dec 12 - 18
Evaluate Model	Dec 19 - 20
Playoff Simulation	Jan 4, '23 - Jan 8, '23
Evaluate Model	Jan 9, '23 - Jan 10, '23
Score Prediction Model	Jan 11, '23 - Feb 2, '23
Evaluate Model	Feb 3, '23 - Feb 5, '23
Betting Simulation	Feb 6, '23 - Feb 12, '23
Evaluate Model	Nov 14 - 15
Presentation Writeup	Feb 25, '23 - Mar 5, '23
Final Report Writeup	Mar 6, '23 - May 1, '23

Deadlines

Specification Draft Deadline	Oct 11
Specification Deadline	Oct 13
Progress Report Soft Deadline	Nov 13
Progress Report Deadline	Nov 16
Presentation Soft Deadline	Mar 1, '23
Presentation	Mar 6, '23 - Mar 17, '23
Final Report Soft Deadline	Apr 25, '23
Final Report Deadline	May 2, '23

Breaks

Christmas	Dec 24, '22 - Jan 1, '23

Figure 5: Original Project Timeline

The majority of extensions were not implemented due to multiple unforeseen circumstances delaying progress - slack was however built into the timeline to accommodate this, ensuring that the key aims of the project were achieved.

Data collection and analysis took longer than expected due to the volume of data required. This meant that more time was needed to create the collection scripts and ensure that missing data was handled without exceptions. Also, the amount of data meant that the execution of the collection took a long time; this increased the chances of Internet connection issues aborting the collection. The Term 1 slack however successfully accounted for this.

Technical issues with personal devices over the Christmas break further delayed progress. As a result, focus was shifted away from implementing the extensions of a Stanley Cup Playoff Predictor and Betting Model, and instead towards maximising the predictive accuracy of models through iterative feature engineering. This also provided data to conclude whether there is an upper limit to predictive accuracy of hockey games due to the influence of luck.

During the data collection phase, it was discovered that the COVID-19 pandemic reduced the number of games played during the 2019/20 and 2020/21 seasons. This led to the idea that that the pandemic may have had an external impact on game outcome. The evaluation of the initial models provided little increase in accuracy compared to previous work, again possibly suggesting some influence from the pandemic. We therefore further investigated this potential impact after the League Simulation instead of implementing the original extensions.

3.3 Risk

There are multiple potential sources of risk associated with this project. The most significant are described here along with action to mitigate them.

3.3.1 Data Availability

The project's success relies on the availability of a wide range of data from every NHL regular season game over the course of multiple seasons. To ensure that the project was viable, the NHL Stats API was immediately explored to confirm that this data was available without the presence of many missing values.

To ensure that this data did not suddenly become unavailable (e.g due to the API being shutdown), the first step of this project was to collect the

data using Python scripts and to make local copies on a personal device.

3.3.2 Data Loss

To prevent the loss of collected data (which would delay the project by having to re-execute the collection scripts), the data was stored in four separate locations: local copies on a laptop, on a USB memory stick, as a Kaggle dataset [23] and on a GitHub repository.

This maximised data redundancy so that multiple points of failure were required for total data loss. All scripts were also stored in these same locations.

3.3.3 Scope Creep

The iterative approach to the project allowed for the scope to change when new ideas became apparent (e.g. the impact of COVID-19 on game prediction). As such, it was possible for the project's goal to significantly shift and the original aim to become obscured.

For this reason, the concrete objective of developing models with improved predictive accuracy, compared to past work, was set with the season simulation being a compulsory feature. Any further objectives that were suggested by the data analysis during this process could then be considered as alternative extensions once this had been completed. These new extensions were evaluated compared to those originally suggested to determine the best course of further work.

4 Data Collection

A pre-existing Kaggle dataset [29] was identified that contained multiple datapoints relating to all regular season NHL games from 2000/01 - 2019/20. In addition to individual game data, team and player information was also provided. All of this data was retrieved from the NHL Stats API [20] which provides live and historical game data for all NHL games.

4.1 NHL Stats API

The NHL Stats API [20] can be accessed and explored by manipulating the URL to return JSON formatted data covering a range of topics.

To access data from a specific game, the Game ID is inserted into the URL. This ID is composed of 10 digits. The first four digits correspond to the season e.g. '2013' for 2013/14. The next 2 digits represent the type of game being played (see Table 3). The final 4 digits are the Game ID. Incrementing from '0001', this indicates the game number of the current season (i.e. the last game of the 2021/22 season being '1312' with 1312 games played that season).

An example of the use of the Game ID is in the URL `https://statsapi.web.nhl.com/api/v1/game/2013020001/feed/live` where '2013020001' is the full Game ID. This request returns JSON formatted data for the first regular season game of the 2013/14 season.

Number	Game Type
01	Preseason
02	Regular Season
03	Playoffs
04	All-star

Table 3: Game ID Digits and Corresponding Game Type

The API stores a large volume of data for each game, the previous example

returning game data that can be used either while the game is live or completed. To improve the readability of this JSON data, an online JSON viewer [30] was utilised which clearly separated the JSON objects into expandable sections.

Due to the lack of documentation available for the API, the data was explored by manipulating the URL based on the public source code of the original dataset collection script [31]. This helped to determine how to access the required data such as team information, player information, overall game stats, penalties etc.

4.2 Python Script

4.2.1 Reasons to Write a New Script

Instead of using the pre-existing dataset, it was decided to write a new Python script inspired by the R program that has been made publicly available along with a discussion of the data [31]. This was for several reasons now discussed.

Firstly, the dataset only contained data from 5th October 2000 to 29th September 2020. It was required to also gather data for the 2020/21 and 2021/22 seasons, therefore the original script would have to have been modified anyway. In addition, it was decided to begin the collection of data for this project from the 2013/14 regular season onwards due to the consistency of the division alignments (excluding the temporary COVID-19 induced realignment [7]) improving the ease of analysing games between selected teams.

Secondly, we have increased proficiency in the Python language compared to R due to more previous experience. Because of time constraints, it was determined that it would be quicker to implement a complete new version

of the script than attempting to learn the intricacies of a new language just to modify some sections of the original script. The increased experience of implementing scripts in Python also allowed for greater confidence in the control of the collection of data and how this was presented/manipulated in the csv files.

Finally, a new script meant that data could be formatted and standardised in a new way. For example, rink coordinates of the plays are not consistent between games due to the fact that the coordinates are relative to where the score-keepers are positioned in the arena. This could mean that the home team is taking shots in the negative x-axis in the first period in one game but in the positive x-axis in another game. This produces a difficulty in analysing period-wise shot positions for a given team due to an increased spread of the data in the coordinate space. A naïve assumption on inspection of this data could result in the conclusion that a team is taking poor shots from within their own half of the rink instead of in the offensive zone. As a result, it was necessary to standardise the coordinates of plays for every game; the home team always attacks in the direction of the positive x-axis in the first period.

4.2.2 Explanation of the Python Script

The script consists of 8 functions that each return csv files related to certain statistical topics e.g. game stats, player stats, goal stats etc. Each function and the files they produce are shown in Table 4.

Function	**CSV File**
`getGameInfo()`	`gameInfo.csv`
`getGameStats()`	`gameStats.csv`
`getTeamInfo()`	`teamInfo.csv`
`getGameOfficialsInfo()`	`gameOfficialsInfo.csv`
`getPlayerInfo()`	`playerInfo.csv`
`getGamePlayerStats()`	`gamePlayerStats.csv`, `gameGoalieStats.csv`
`getPlaysStats()`	`playsStats.csv`, `goals.csv`, `penalties.csv`, `playPlayersStats.csv`
`getShifts()`	`shifts.csv`

Table 4: Functions to Collect NHL Game Data

Data was only collected for the regular season games. This was decided due to the amount of repetition in the best-of-seven playoff series which might impact the style of play for certain teams compared to League games. For example, in the earlier ties, teams may be less offensive than usual to mitigate the risk of being scored on near the start. As a result, it may instead be best to collect playoff data in a separate database and to train models for this separately.

Alternatively, it would be possible to test the models trained on regular season data to investigate whether there truly is a difference in the style of play in playoffs; this would be concluded if the accuracy score of these predictions are significantly different to that of regular season predictions.

The functions were required to handle the presence of missing data. This was implemented with the use of `try` and `except` blocks for each value accessed in the JSON data. If a `KeyError` is thrown, then the missing value is represented by 'NA' in the dataset.

It was also necessary to ensure that only data from completed games was

collected. This is an issue in the 2019/20 season which was curtailed due to the outbreak of the COVID-19 pandemic [32]. As such, the final few games of this season were not played; however, the NHL still records these games as 0-0 draws with no shots by either team (and other similarly even stats).

Including these games may skew the data and, therefore, predictions made by the models, hence they must be omitted from the dataset. This was achieved by limiting the Game ID in the 2019/20 season to 1082 (i.e the number of games played).

To construct the csv files, each function was executed separately. Due to the volume of data collected, on average this required between 30 minutes to 2 hours per function. Upon completion, the files were opened in Excel [22] to quickly evaluate that the required data was successfully collected from the API.

4.2.3 Collected Data

The following tables detail the attributes of the data stored in each csv file. These files were then uploaded to form a new Kaggle dataset [23]. From this, feature vectors could be created by calculating new statistics from the data; this process is discussed further in *Machine Learning Models*.

Brief justifications for why such data is collected are given before a breakdown of each subset. These are explained further, and given extra context, in the *Data Analysis* section.

Game Information

Brief overview of every game (10,724 entries).

Since the goal of the models is to predict game outcome, it must be known which games occur when and between which teams. A location feature is also required which determines the home and away team in order to explore

whether travelling has an impact on the result. In addition, the ground truth of the game winner is determined as the team that has scored the most goals. This ground truth is required for both training the models and evaluating the predictive accuracy on the test set.

Table 5 shows the data collected to satisfy these requirements.

Attribute	**Description**
gameId	Unique ID of the game
date	Date the game was played
homeTeam	Unique ID of the home team
awayTeam	Unique ID of the away team
homeGoals	Number of goals scored by the home team
awayGoals	Number of goals scored by the away team
result	Game ended in Regulation ('REG') or Overtime ('OT')
homeRinkSide	Side of the rink that the home team's bench is located. Used to standardise all coordinates

Table 5: Game Information Data

Team Information

Every team that has played in 2013/14 - 2021/22 season (32 entries).

Within the new dataset, teams are identified by their unique Team ID. As such, a mapping to the actual team name is required to easily recognise teams when results are output to the user. This data is shown in Table 6.

Attribute	**Description**
teamId	Unique ID of the team
name	Long name of the team e.g. 'New York Rangers'

Table 6: Team Information Data

Game Officials Information

Information on the four on-ice game officials - usually 2 referees, 2 linesmen (42,887 entries).

Table 7 shows how data on the officials of each game has been collected. This opens up the possibility of exploring whether there's a correlation between a set of officials and game outcome.

Although it is unlikely that there is a direct influence, if a certain official tends to call more penalties, then a team that often concedes on a penalty kill may be scored on more this game. This team would therefore have an even lower chance of winning the game than usual.

Attribute	**Description**
gameId	Unique ID of the game
name	Full name of the official
position	Either 'Referee' or 'Linesman'

Table 7: Game Official Information Data

Player Information

Information on every player that played at least one game between 2013/14 - 2021/22 (2,270 entries).

Each player is identified by a unique Player ID. For readability, it is required to also know the player name. This is useful for analysing a potential

player correlation metric which determines the performance of a certain line of players in a team. This is further discussed in the *Shift Statistics* data.

The collected data which may help to implement such a metric is displayed in Table 8.

Attribute	**Description**
playerId	Unique ID of the player
fullName	Player's full name
nationality	Player's nationality
position	Playing position: 'D' = Defence, 'L' = Left Wing, 'R'= Right Wing 'C' = Centre 'G' = Goalie
birthdate	Player's date of birth (YYYY-MM-DD)
height	Player's height (ft' inches")
weight	Player's weight (lbs)
shootsCatches	The side a skater shoots on or a goalie catches with ('L', 'R')

Table 8: Player Information Data

Game Statistics

More detailed statistics from every game between 2013/14 - 2021/22 (10,724 entries).

More statistics are required from each game other than just the final score and teams involved. For example, the number of shots on goal a team takes can be used to determine the percentage of shots that led to goals. The number of powerplays a team has can help determine game outcome if a team tends to score more goals with the player advantage. A team blocking more opposing shots may indicate a stronger defence. All of these correlations need to be analysed further, therefore a wide range of game statistics for both the home and away team are collected.

These examples are contained within Table 9. Note that the same statistics are recorded for both the home and away team of each game.

Attribute	**Description**
gameId	Unique ID of the game
homeShots	Number of home team shots on goal
homeHits	Number of hits inflicted by the home team
homePim	Number of penalty minutes served by the home team
homePowerPlayOpportunities	Number of powerplays for the home team
homePowerPlayGoals	Number of powerplay goals scored by the home team
homeFaceOffWinPercentage	Percentage of faceoffs where the home team gains possession
homeGiveaways	Number of occurrences when a home player's action results in loss of puck possession
homeTakeaways	Number of occurrences when the home team is defending and a home player is the direct cause of a gain in puck possession
homeBlocked	Number of shots by the home team that are blocked
awayShots	Number of away team shots on goal
awayHits	Number of hits inflicted by the away team

Continued on next page

awayPim	Number of penalty minutes served by the away team
awayPowerPlayOpportunities	Number of powerplays for the away team
awayPowerPlayGoals	Number of powerplay goals scored by the away team
awayFaceOffWinPercentage	Percentage of faceoffs where the away team gains possession
awayGiveaways	Number of occurrences when an away player's action results in loss of puck possession
awayTakeaways	Number of occurrences when the away team is defending and an away player is the direct cause of a gain in puck possession
awayBlocked	Number of shots by the away team that are blocked

Table 9: Game Statistics Data

Player Game Statistics

Game statistics on a player-by-player basis - skaters only (385,885 entries).

The game statistics are further advanced by considering individual player performance in Table 10. These individual statistics can be used to determine a player correlation metric explained within the *Shift Statistics* breakdown.

This metric could be used to produce a model that outputs an expected number of goals produced by a set of players as input. For example, a player with an increased number of powerplay goals and increased time on ice during

a powerplay is more likely to score a goal than on average.

Each statistic collected in this data subset reflects the team-wide statistics collected in the *Game Statistics*.

Attribute	**Description**
gameId	Unique ID of the game
playerId	Unique ID of the player
teamId	Unique ID of the player's team
timeOnIce	Amount of time in seconds spent on the ice by the player
assists	Number of assists for a goal provided by the player this game (usually last 2 passes before a goal)
goals	Number of goals scored by the player this game
shots	Number of shots taken by the player this game
hits	Number of hits this player inflicted on other players this game
powerPlayGoals	Number of powerplay goals scored by the player this game
powerPlayAssists	Number of assists for a powerplay goal provided by the player this game
penaltyMinutes	Number of minutes the player served in the penalty box this game
faceOffPct	Player's faceoff win percentage ('N/A' if no faceoffs taken) this game
faceOffWins	Number of faceoffs won by the player this game

Continued on next page

faceoffTaken	Number of faceoffs taken by the player this game
takeaways	Number of individual takeaways by the player this game
giveaways	Number of individual giveaways by the player this game
shortHandedGoals	Number of penalty kill goals scored by the player this game
shortHandedAssists	Number of assists for a penalty kill goal provided by the player this game
blocked	Number of shots taken by the player that are blocked this game
plusMinus	+1 if their team scores an even-strength/shorthanded goal while on the ice, -1 if opposing team scores an even-strength/shorthanded goal while on the ice this game
evenTimeOnIce	Amount of time the player spent on the ice while 5-on-5 this game
powerPlayTimeOnIce	Amount of time the player spent on the ice while on a powerplay this game
shortHandedTimeOnIce	Amount of time the player spent on the ice while on a penalty kill this game

Table 10: Player Game Statistics Data

Goalie Game Statistics

Game statistics on a player-by-player basis - goalies only (230,49 entries).

Similar to the *Player Game Statistics*, the team game stats can be refined to consider the goalie's performance. It can be argued that the goalie's actions are one of the most important factors to consider for determining game outcome since a strong performance can keep a weak team in a game, and a poor performance can prevent a strong team from winning.

As such, it is necessary to determine the goalie's save percentage - the percentage of shots saved per game. This can further be refined to consider the strength of the opposing team at the time of the shot. For example, having a higher save percentage while the opposition is on a powerplay can help negate the advantage of the attacking team. These statistics are all shown in Table 11.

Attribute	**Description**
gameId	Unique ID of the game
playerId	Unique ID of the goalie
teamId	Unique ID of the goalie's team
timeOnIce	Amount of time in seconds spent on the ice by the goalie
assists	Number of assists for a goal provided by the goalie (usually last 2 passes before goal)
goals	Number of goals scored by the goalie this game
pim	Number of penalty minutes served (will be served by a skater) this game
shots	Number of shots faced this game
saves	Number of shots saved this game
powerPlaySaves	Number of shots saved while on a penalty kill this game
shortHandedSaves	Number of shots saved while on a powerplay this game
evenSaves	Number of shots saved while at even strength this game
shortHandedShotsAgainst	Number of opposition shots faced while on a powerplay this game
evenShotsAgainst	Number of opposition shots faced while at even strength this game
powerPlayShotsAgainst	Number of opposition shots faced while on a penalty kill this game
decision	Win ('W') / Lose ('L') game
savePercentage	Percentage of all shots saved this game
powerPlaySavePercentage	Percentage of all shots saved while on a penalty kill this game
evenStrengthSavePercentage	Percentage of all shots saved while at even strength this game

Table 11: Goalie Game Statistics Data

Play Statistics

Statistics on every play between 2013/14 - 2021/22 (3,428,494 entries).

Each significant action on the ice such as a shot, goal, penalty or stoppage is recorded as a unique 'play'. Further explanation of the play is also provided such as the type of shot or the reason for the stoppage. This is useful for analysing the impact of different shot types - discussed in the *Data Analysis*.

Another use of this data is the location of all plays on the rink are also recorded. This allows the construction of heatmaps for each type of play that indicate regions of higher activity. From this, it could be possible to build a location-based expected goals metric where the location of the shot on the ice informs the probability of scoring a goal.

All data collected for each play is described in Table 12.

Attribute	**Description**
playId	Formed by 'gameId_playNumber' (incremental from 0)
gameId	Final 4-digits of the unique game ID (i.e. game number of the season)
teamIdFor	Unique ID of the team the play is for
teamIdAgainst	Unique ID of the team the play is against
	Continued on next page

event	Type of play, either: 'Game Scheduled', 'Period Ready', 'Period Start', 'Faceoff', 'Hit', 'Stoppage', 'Shot', 'Takeaway', 'Blocked Shot', 'Missed Shot', 'Giveaway', 'Period End', 'Period Official', 'Goal', 'Penalty', 'Game End', 'Game Official', 'Official Challenge', 'Shootout Complete', 'Early Intermission Start', 'Early Intermission End', 'Emergency Goaltender'
secondaryType	Optional extra detail about the play e.g. 'Wrist shot', 'Slap Shot', 'Aggressor', 'Broken Stick'
description	Brief written description of the event
x	Recorded x-coordinate of the event on the rink
y	Recorded y-coordinate of the event on the rink
period	Which period the play was in (1, 2, 3, 4 = OT, 5 = SO)
periodType	'REGULAR', 'OVERTIME', 'SHOOTOUT'
periodTime	Time in seconds from the start of the current period
periodTimeRemaining	Time in seconds left in the current period

Continued on next page

dateTime	Real datetime of the play
homeGoals	Number of home goals in the game at the moment of the play
awayGoals	Number of away goals in the game at the moment of the play
xStandardised	Standardised x-coordinate of the event on the rink
yStandardised	Standardised y-coordinate of the event on the rink

Table 12: Play Statistics Data

Goal Statistics

Statistics on every goal between 2013/14 - 2021/22 (63,300 entries).

Data from every goal is required to determine the strength of the team at the time of scoring. This allows us to calculate the percentage of powerplays and penalty kills a team scores on or successfully defends respectively. As will be discussed in the *Data Analysis*, team strength is a considerable factor to consider when determining game outcome.

Table 13 shows all data collected for each goal.

Attribute	**Description**
playId	Unique ID of the play recording the goal
gameWinningGoal	Was this the goal that led to the team winning? ('True'/'False')
emptyNet	Was there no goalie in the goal? ('True'/'False')
strength	'EVEN', 'PPG', 'SHG'

Table 13: Goal Statistics Data

Penalty Statistics

Statistics on every penalty served between 2013/14 - 2021/22 (81,897 entries).

To determine the strength of a team at any given point of play, it must be known whether any penalties are currently being served. By using the ID of the play the penalty was issued on, this can be determined by comparing the current game time to the amount of time the penalty covered. This is made possible by the data collected in Table 14.

Knowing the current strength of a team is useful for calculating puck possession statistics such as 'Corsi' which measures the proportion of shots a team takes while at even strength.

Attribute	**Description**
playId	Unique ID of the play recording the penalty
penaltySeverity	'Minor', 'Bench Minor', 'Major', 'Misconduct', 'Match', 'Penalty Shot', 'Game Misconduct'
penaltyMinutes	2 (minor), 5 (major), 10 (misconduct), 4 (double minor), 0 (penalty shot)

Table 14: Penalty Statistics Data

Player Play Statistics

Statistics on every play on a player-by-player basis (5,776,002 entries).

The play data can be further refined by identifying the players involved in the play. Similar to other collected subsets, this would be useful for constructing a player correlation metric. For example, the number of shots or goals each player makes can be calculated to determine the conversion rate of an individual player. A player with a higher rate that also has more ice time should lead to more goals being scored for the team.

The data required to identify individual players involved in plays is shown in Table 15.

Attribute	Description
playId	Unique ID of the play the player was involved in
id	Unique ID of the game the play occurred in
playerId	Unique ID of the player
playerType	'Winner', 'Loser', 'Hitter', 'Hittee', 'Shooter', 'Goalie', 'PlayerID', 'Blocker', 'Unknown', 'Scorer', 'Assist', 'PenaltyOn', 'DrewBy', 'ServedBy'

Table 15: Player Play Statistics Data

Shift Statistics

Statistics of every shift in every game on a player-by-player basis (8,801,041 entries).

Due to the use of rolling substitutions throughout a game, team performance may be varied depending on the players currently on the ice. A player correlation metric can therefore be constructed that determines the performance of a line of players by measuring the percentage of goals they score for the team. If a line has a higher scoring percentage and is played more often, then the team should in turn have a greater probability of winning.

To determine how often a line is played, it must be known when each player performs a shift on the ice. The length of time when all players in a given line are simultaneously on the ice can be used to predict the likelihood of a goal being scored.

Table 16 shows the data available relating to a player's shifts.

Attribute	**Description**
gameId	Unique ID of the game the shift occurred in
playerId	Unique ID of the player
period	Period number the shift occurred in
startTime	Time from the start of the current period in seconds at the start of the shift
endTime	Time from the start of the current period in seconds at the end of the shift

Table 16: Shift Statistics Data

5 Data Analysis

Since the approach to solving the problem is to use supervised machine learning techniques, selecting the most relevant features to feed into the models is a manual task. This is achieved through analysing trends in the collected data to identify any significant patterns that could be of use to maximise the models' predictive ability.

All graphs discussed in this section were produced using the Python matplotlib plotting library [18]. This is a popular library used to visualise data for improved analysis. With the use of this library, several graphs were produced from the collected data such as bar charts, line graphs and heatmaps.

These graphs were created by importing the chosen csv files stored in the Kaggle dataset [23] into Pandas Dataframes [17]. The data required to be viewed could then be selected from these Dataframes to be used as the input data for the relevant matplotlib methods to construct the graphs. These graphs are discussed in detail in the relevant subsections that follow.

5.1 Games per Season

The first notable feature of the data is that there is an increasing number of games played over the 9 recorded seasons as more teams are added to the League. 1230 games per season were played from 2013/14 - 2016/17. This increased to 1271 games per season between 2017/18 - 2018/19 with the addition of the Vegas Golden Knights [8] followed by a further increase to 1312 games per season in 2021/22 with the arrival of the Seattle Kraken [9]. As a result, the general trend of goals scored per season is also increasing (Figure 6). More games results in more time to score goals therefore this is expected and not indicative of a change in the style of play over time. This could however provide issues when predicting game outcomes between seasons.

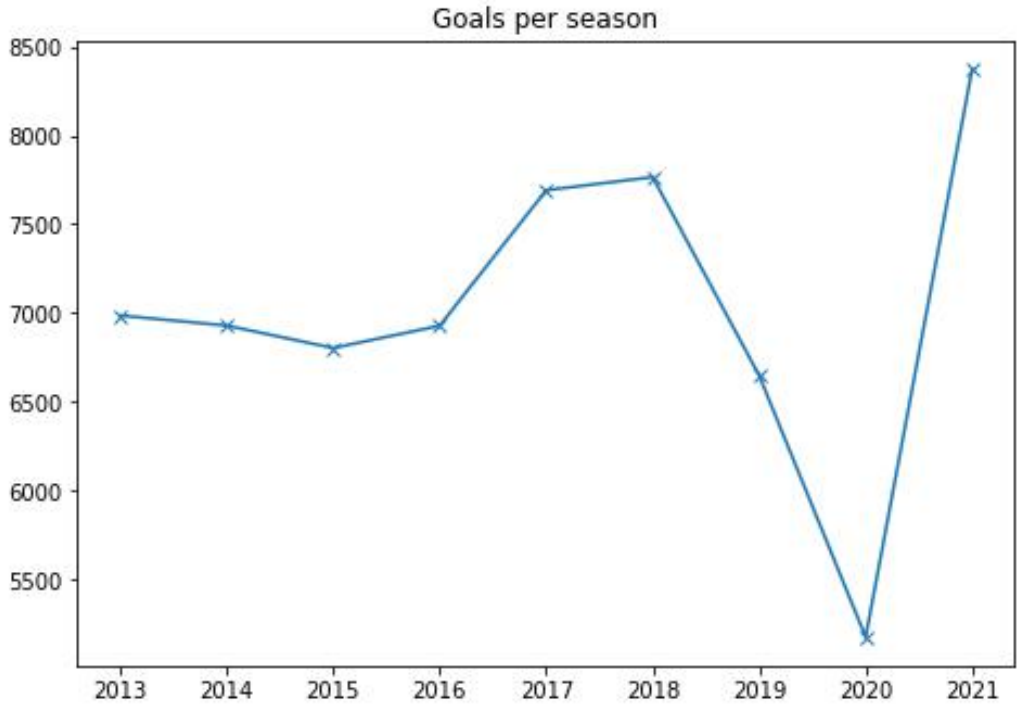

Figure 6: Total Number of Goals Scored per Season

The COVID-19 pandemic led to a curtailing of the 2019/20 season on March 12th 2020 [32] in an attempt to mitigate the spread of the disease. This meant that only 1082 games were played that season.

The continuation of the pandemic into the 2020/21 season also resulted in a reduced number of games played, along with a temporary division realignment due to travel restrictions [7]. This led to a season consisting of 868 games.

As a result, the number of goals scored during these two seasons also significantly reduced in comparison to other 'normal' seasons. The impact of this deviation in the general data trend on the predictive power of the models is explored in *Investigating the Impact of the COVID-19 Pandemic on Game Prediction*.

Overall, data was collected for 10,724 games over the course of the 9

seasons.

5.2 Team Strength

Figure 7 displays the total number of goals scored each season, separated by the strength of the scoring team, where 'EVEN' means even strength (no penalty or equal number of skaters on either team), 'PPG' is scoring on a powerplay and 'SHG' is scoring on a penalty kill.

It can be seen that even strength goals vastly and consistently outweigh the other two groups in number of goals. This is intuitive since penalties are rare events in comparison to total game time therefore most of the play will occur at even strength. The larger amount of time played in this state therefore leads to more goals being scored during this time.

The second most number of goals come when the team is on the power-play. This again makes sense since having a player advantage should result in the attacking team with an unmarked player on the ice who has both the time and space to set up in a good position to maximise scoring opportunities.

A significantly lower proportion of goals are scored by the shorthanded team. This is due to the team being overwhelmed by the opposition with the extra skater, meaning that they are often held back in their defensive zone. It is therefore harder to maintain possession of the puck for long enough to initiate a breakout into the offensive zone for any chance of scoring. Through observation of hockey games, teams often score shorthanded due to lucky breaks in play or catching the opposition off-guard during line changes.

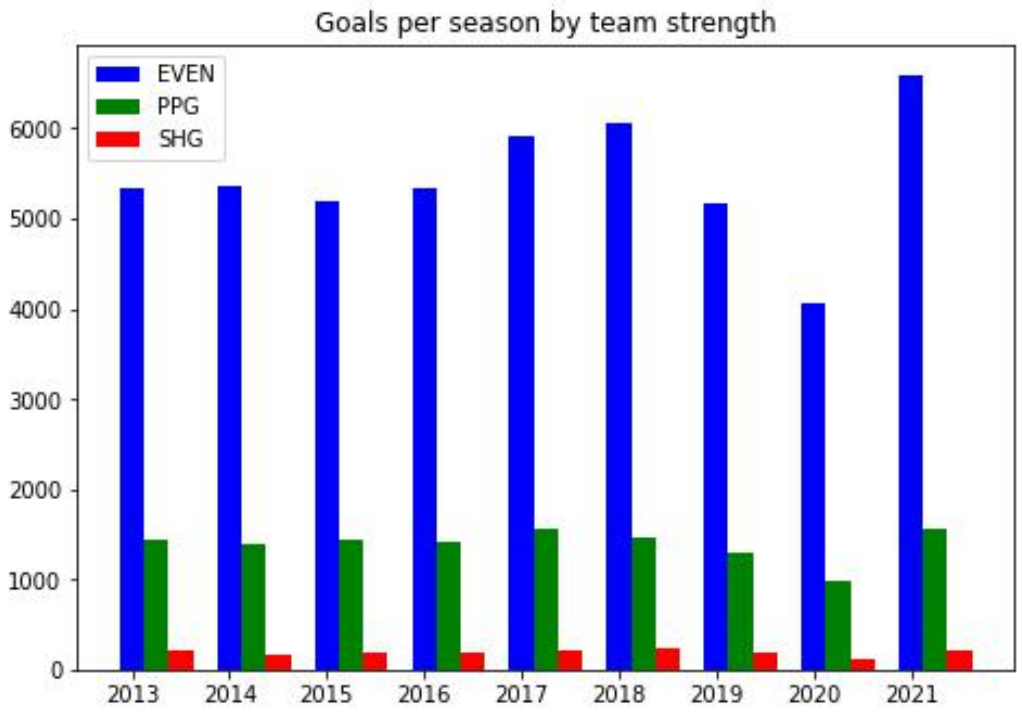

Figure 7: Goals Scored per Season by Scoring Team Strength

This suggests that team strength is a good indicator for the number of goals scored in a game. A team that spends more time on the powerplay should also score more goals and the opposite for increased time shorthanded.

5.3 Shot Outcomes

Each shot can have one of four outcomes: saved, goal, blocked, missed. A shot is saved if it is either caught and held by the goalie (causing the referee to stop the play and restart with a faceoff) or deflected away by the goalie. A shot is blocked if the puck is stopped by a defender, therefore never reaching the goal. A missed shot is one that makes it all the way through the defense but is not on target to the goal.

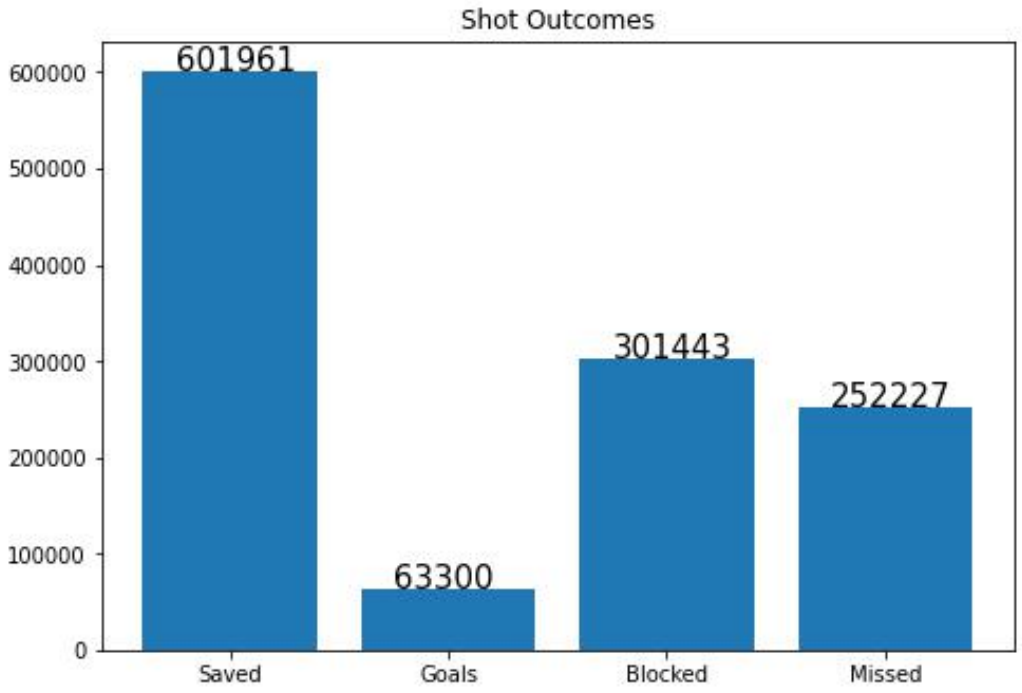

Figure 8: Goal Shot Outcomes

Figure 8 displays the number of each type of shot recorded. It is clear that goals (63,300) are significantly rare events when compared to the total number of shots taken (1,218,931). An interesting point is that 49.4% of shots are on target and require the goalie to make a save. When including the number of goals, this means that 54.6% of all shots have a chance of scoring. This corresponds to the relatively high scores often observed at the conclusion of hockey games and also encourages attackers to take more speculative shots due to the high chance of challenging the goalie.

This also highlights the importance of the goalie throughout the game and suggests that goalie statistics are the most important individual player stats to consider when determining the game outcome. A goalie with a relatively lower percentage of shots saved would be significantly detrimental to the team's performance due to the high frequency of opposition shots faced.

The second most number of shot outcomes are blocked shots (24.7%). This also highlights the importance of defenders mitigating the danger of shots by placing their body in front of the puck. Due to the importance of blocked shots, a team with a lower proportion of blocked shots in a game compared to the total number of shots taken may suggest a weaker defence and more shots reaching the goal (therefore increasing the probability of scoring). This statistic would however have to be used in conjunction with the proportion of missed shots since only shots of a higher quality (i.e. on target) impact the number of goals scored. A team may have fewer blocked shots only because more of their shots miss the target, therefore do not need to be blocked anyway.

The different shot outcomes provide an insight into how different statistics may be correlated to each other. This correlation also results in the need for a large number of features to be collected to truly capture the context of a game and maximise goal prediction accuracy.

5.4 Shot Type

A player has multiple types of shot to take depending on the game situation (see Table 17). The number of certain shots a team takes during a game could indicate their attacking dominance.

Shot Type	**Explanation**
Wrist Shot	The player faces the goal with the puck to the side of the body. The puck is dragged forward along the ice with a final flick motion from the wrist for power. Taken from most locations on the ice.
Slap Shot	More similar to a golf swing, the puck is hit, rather than dragged, for increased power but generally decreased accuracy. Often taken from further distances from the goal.
Backhand	Shot taken on the convex side of the stick blade. Usually harder to direct and has less power therefore taken close to the goal.
Snap Shot	Similar to a wrist shot but the puck is dragged for a shorter distance before quickly snapping the stick forward for increased power. More body weight is also placed into the stick shaft for larger flex.
Tip-in	The attacking player reaches their stick out to slightly change the direction of the puck already travelling towards the goal. Often taken in front of the goal.
Deflected	The defender makes accidental contact with the puck, causing it to change direction and score in their own net. Often from in front of the goal.
Wrap-around	The attacker skates towards one side of the goal. Instead of shooting, they carry on around the back of the net before tightly placing the puck into the goal by the opposite goal post. Can only be scored by the side of the net.

Table 17: Explanation of Shot Types

Wrist shots are often taken from most locations on the ice due to the versatility of power that can be applied and its good accuracy. Considering the number of these shots taken only indicates the number of potential scoring opportunities a team has and does not provide much deeper information.

Due to slap shots often being taken from further away from the goal, an increase in the number of these shots may indicate that although the team is being dominant offensively, they are also being matched by an opposition with a strong defence, forcing the play back towards the blue line. A strong defence would in turn concede fewer goals. In addition, more of these shots could suggest that the attacking team is becoming more desperate in their attempts to score, as they are taking shots deemed to be less accurate. A team would only become desperate when either losing or drawing a game, therefore this may be a good flag of a tight or losing game.

The number of these shots taken alone provides little information on game prediction without knowing how many of these shots actually result in goals. Figure 9 displays the number of goals scored between 2013/14 - 2021/22 grouped by shot type.

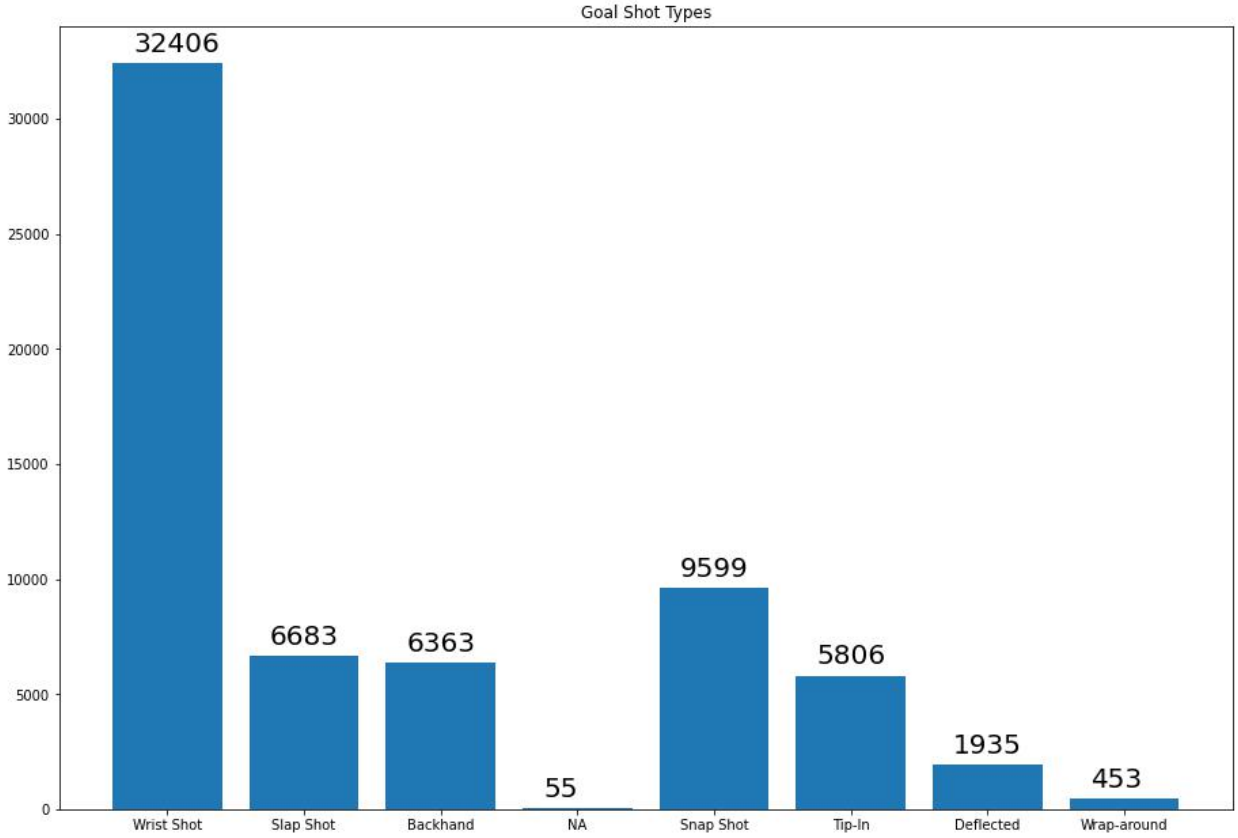

Figure 9: Goal Shot Types

From this, it can be seen that most goals are scored from a wrist shot (32,406), due to its aforementioned versatility, and the least scored by a wrap-around (453) due to its difficulty in terms of skill and the ability of the goalie to quickly cover both goal posts.

With this data, it is possible to define a conversion rate for each shot type that captures the percentage of shots of this type that lead to goals:

$$ConversionRate(x) = \frac{NumberGoals(x)}{TotalShots(x)}$$

Where x is a certain shot type. The calculated conversion rate of each recorded shot type is shown in Table 18.

Shot Type	**Conversion rate**
Wrist Shot	9.09%
Slap Shot	6.19%
Backhand	11.8%
Snap Shot	9.98%
Tip-in	17.8%
Deflected	17.7%
Wrap-around	6.62%

Table 18: Conversion Rate of Each Shot Type

Shots with a higher conversion rate are those that are more likely to result in a goal. This data supports the hypothesis that slap shots result in fewer goals, with a conversion rate of only 6.19%. A team that takes more of these shots in proportion to other shots is therefore likely to score fewer goals, thus having a lower chance of winning the game. This again supports the idea that a team taking more slap shots may be indicative of a strong opposition defence.

Deflections and tip-ins have significantly higher conversion rates compared to the other shot types, with 17.7% and 17.8% respectively. As previously mentioned, this is likely a result of the goalie not having adequate time to react to the small changes in the direction of the puck due to the high speed of the puck's movement. A team that achieves more of these shots is therefore more likely to score more of goals.

There is however a slight issue with this generalised idea. Although tip-ins are deliberate actions and can be set up with a pass across the face of the net, deflections are often more accidental due to them being caused by the defender - it is highly improbable that a defender would purposefully use their body to score an own goal.

Deflections can however be forced by the attacking team by frequently taking powerful shots from distance while there are many players crowding

the front of the net. This is because an increase in distance travelled by the puck increases the chance of the shot coming into contact with another player; the increase in shot speed also reduces the amount of time players have to move out of the puck's trajectory, again increasing the probability of contact.

This description fits the definition of the slap shot, therefore many deflections may be the result of these shots. It is therefore not as clear-cut to conclude that a team taking more slap shots will score fewer goals due to the inverse conclusion that deflections result in the highest number of goals and the hypothesis that more slap shots could result in more deflections.

Unfortunately, only one shot type is recorded by the NHL therefore it is not possible to test this hypothesis as it is not known what type of shot/pass immediately preceded the shot. We can therefore only assume that the team scoring with more deflections/tip-ins only are likely to score more goals.

5.5 Shot Location

Shots can be taken from any location on the ice, with most being taken within a team's offensive zone (i.e. past the blue line) - see Figures 10 and 11. This is mainly due to the offside rule enforcing that the puck must cross the offensive blue line before any attacking player. As a result, the offensive team often attempts to 'gain the blue line' by ensuring that the majority of players are within the offensive zone before setting up a shot attempt; this allows for the defenders on the attacking team to keep any deflected pucks within the zone.

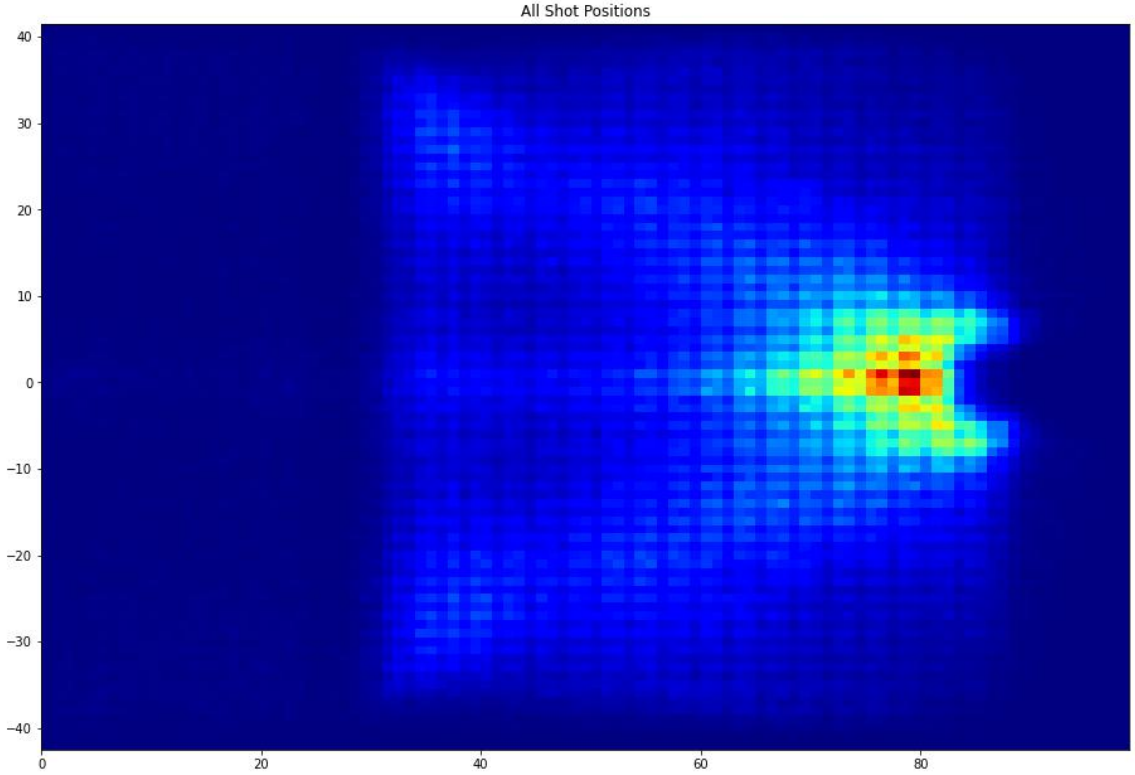

Figure 10: All Shot Locations

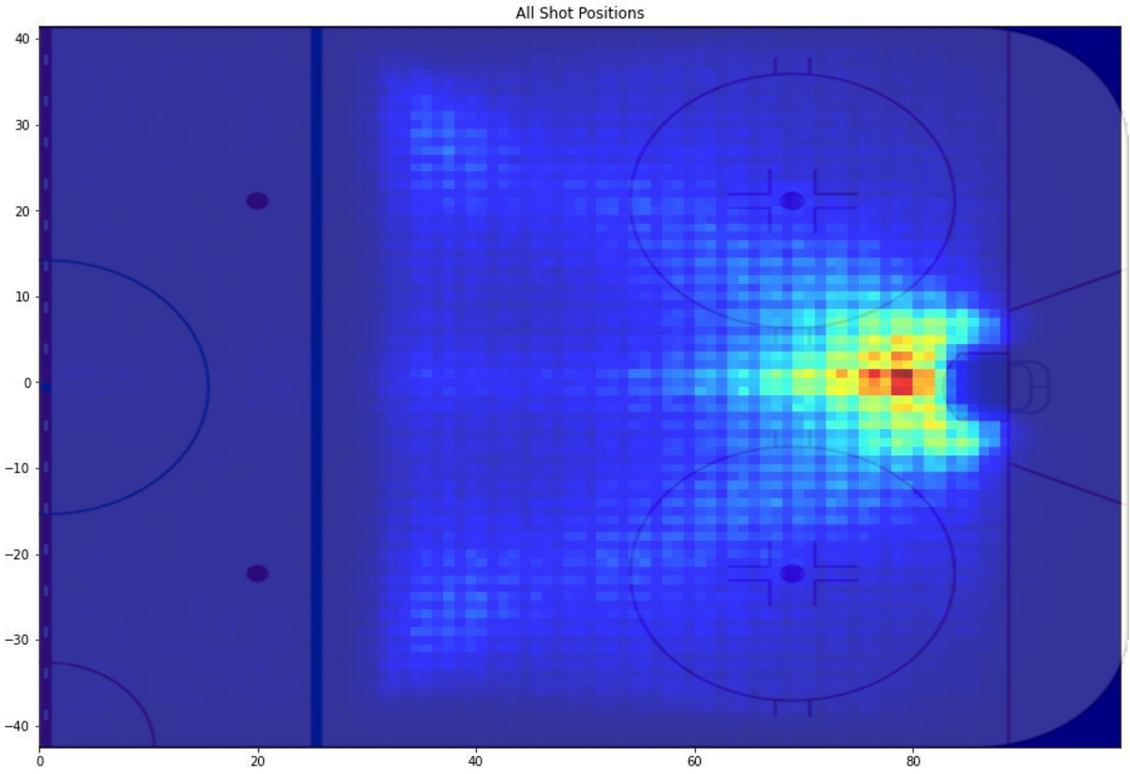

Figure 11: All Shot Locations with Rink Superimposed

Heatmaps have been created from the collected shot data which visualise the number of shots taken from certain locations on the rink with the use of the matplotlib `hist2d()` method. Using the standardised coordinates, all shots have been mapped to the positive x-axis. This is necessary in order to analyse all shots taken around the goal - it does not matter which end of the rink the shot was taken in since this is a mirror image of the opposite side. This provides easier analysis of where exactly most shots were taken from over the course of the nine seasons.

Although it is possible to take a shot from the attacking team's half of the ice, these occurrences are rare (often when shooting on an empty net towards the end of the game) therefore do not skew the data to a noticeable amount and can simply be ignored.

Each pixel of the heatmap represents 1 square foot on the ice. Blue regions indicate fewer shots being taken from this location with red regions indicating the most. For each heatmap, a version with the hockey rink superimposed has also been provided to help understand where exactly these locations are, in reference to the playing area.

Figure 11 shows how most shots are taken at close range and directly in front of the goal. This corresponds to the Centre player position who's general role is to score the most goals. The locations of these shots then fan out away from the goal, following the path of three channels. One channel is formed perpendicular to the goal face down the centre of the ice. This again follows the general area occupied by the Centre.

The other two channels are directed towards the two offensive zone faceoff circles towards the edge of the rink. This corresponds to regions occupied by the Left Wing and Right Wing respectively. These players often take shots at tighter angles to the goal, aiming to find gaps that the goalie cannot cover.

Shot frequency then decreases with an increase in distance before again peaking near the blue line around either side of the rink's width. This is where the defenders are often positioned 'at the point'. Here, defenders can set up for a slap shot with the forwards passing the puck back for a one-touch hit.

The clear boundaries formed by these shots may suggest that it is possible to implement a location-based xG metric similar to that used in football [2].

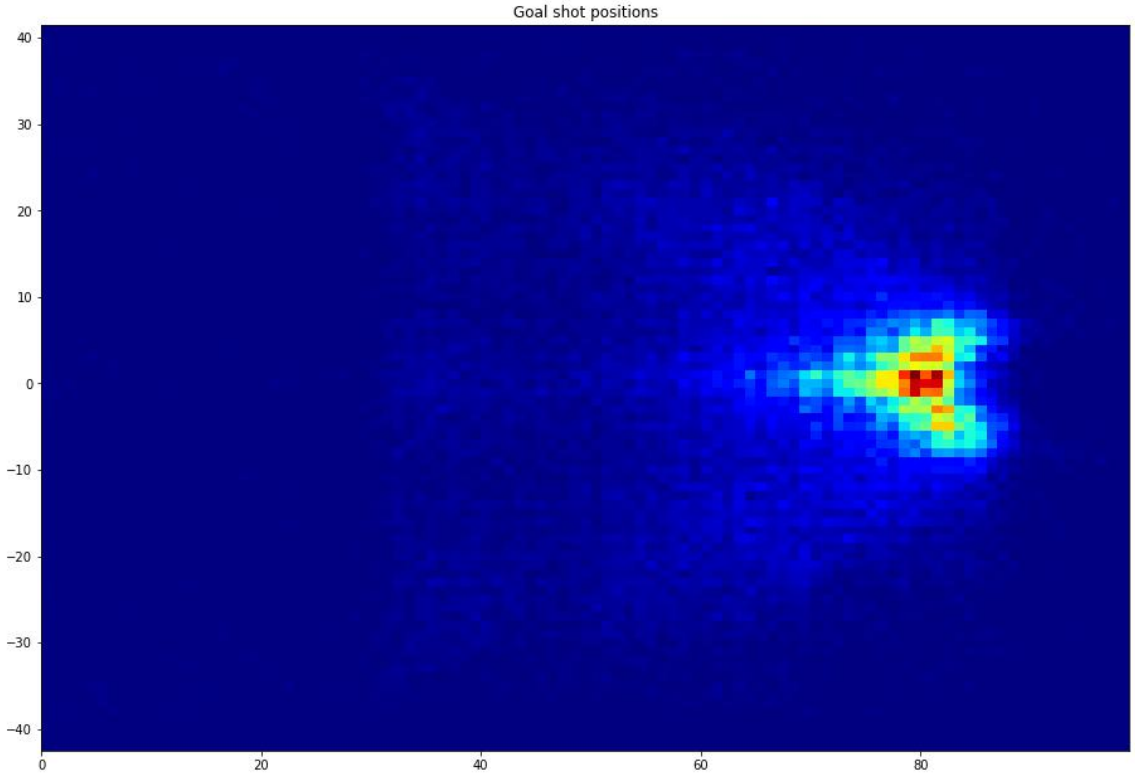

Figure 12: All Goal Locations

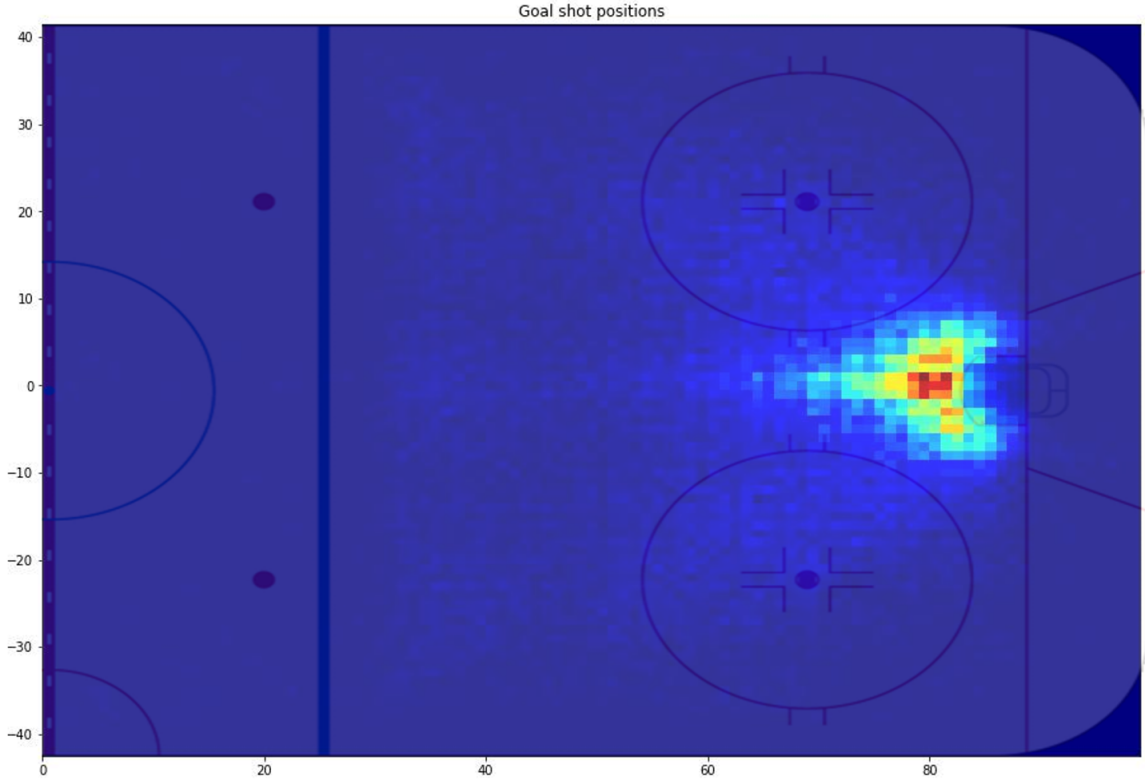

Figure 13: All Goal Locations with Rink Superimposed

Figures 12 and 13 support the idea of a location-based xG metric where these heatmaps indicate the locations of shots that result in goals. Similar to football, most goals are scored close to the goal and at larger angles to the face of the net. The number of goals then appears to decrease with an increase in distance and a decrease in angle. This is likely due to similar reasons as in football: shots closer to the net give the goalie less time to react with a save and larger angles provide the shooter with more areas not covered by the goalie.

Shooting from a larger angle to goal mouth is even more influential in hockey due to the decreased goal size. The dimensions of a NHL goal are 1.8m x 1.2m (6ft x 4ft). Goalies wear heavily padded equipment, meaning that they

can cover most of the goal mouth just by being positioned in the centre of the goal.

As such, it is necessary for the shooter to try and take shots that maximise the number of open areas (i.e. directly in front of the centre of the goal). This is reflected in the heatmaps with a slightly elongated area directly in this central area that does not experience a decrease in shot frequency with an increase in distance at the same rate as other angles.

There is also a minor increase in goals scored at locations occupied by the attacking defenders near the blue line. This corresponds to goals scored by defenders, often through slap shots. This local maxima also adds a complexity to the location-based xG metric that is not as prevalent in football. It is therefore not possible to implement the exact same theory into a hockey xG metric due to multiple locations on the rink providing a varying goal frequency.

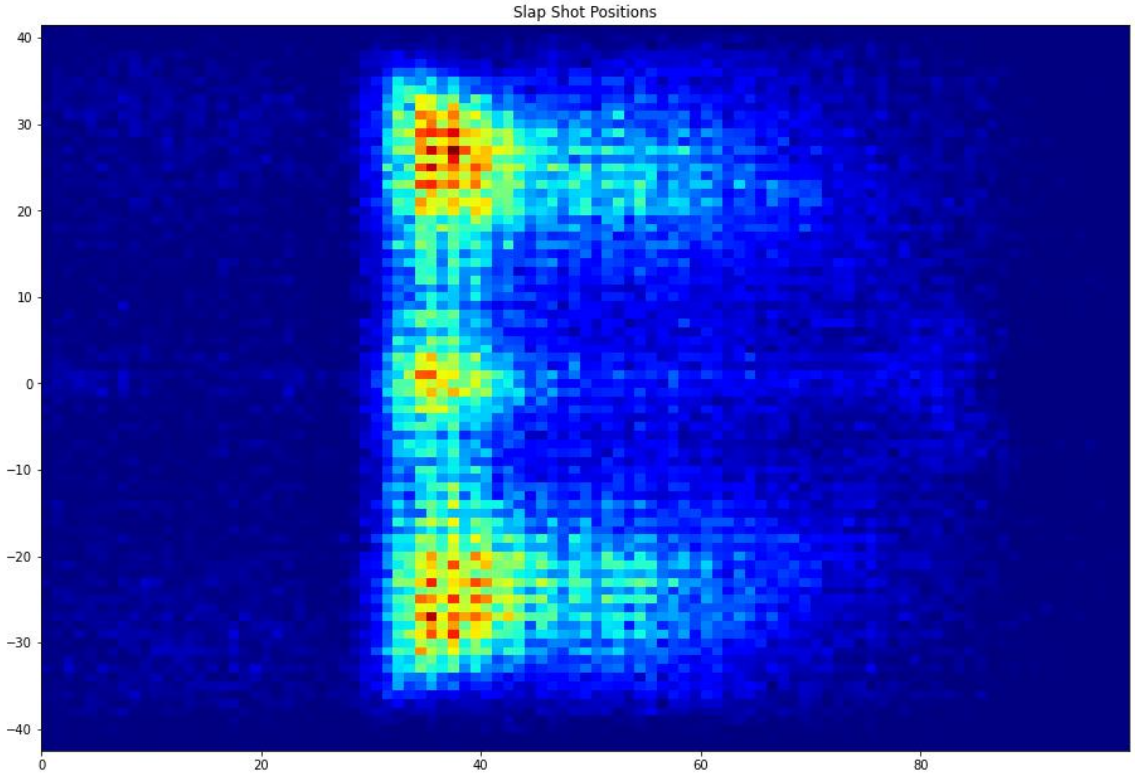

Figure 14: All Slap Shot Locations

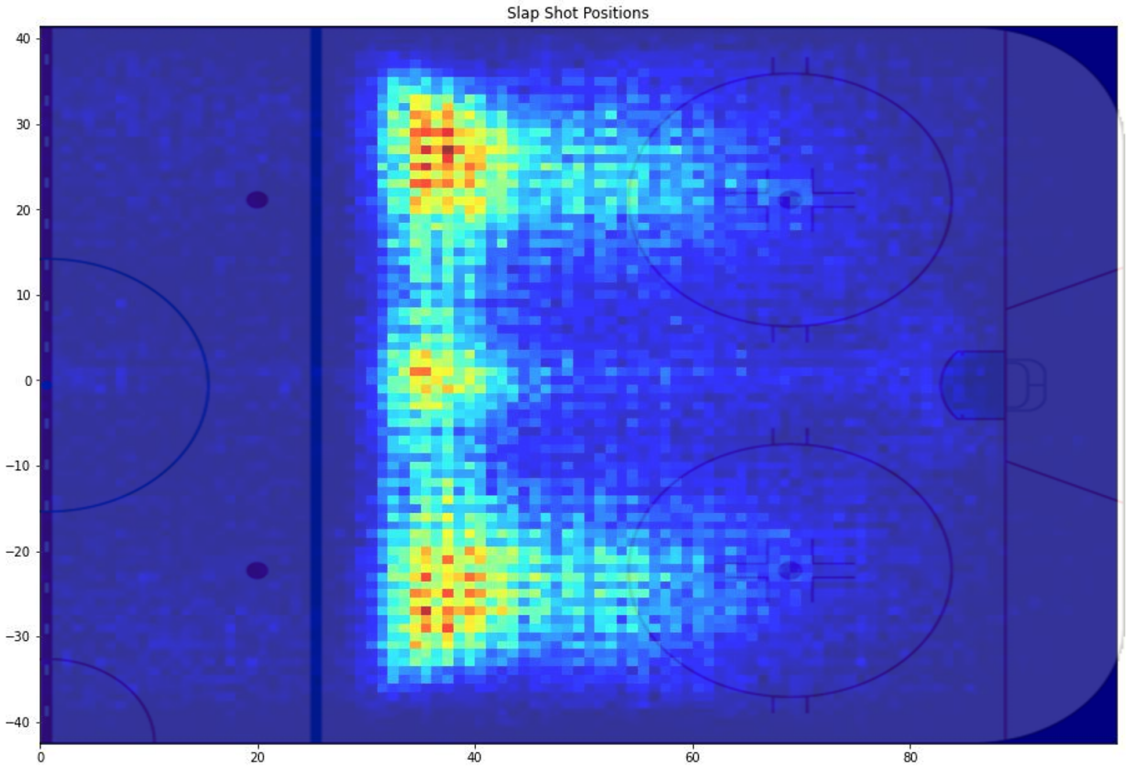

Figure 15: All Slap Shot Locations with Rink Superimposed

Figures 14 and 15 further highlight how the slap shot is taken more frequently at increased distances to goal, with three major regions located along the blue line of the offensive zone. As mentioned previously, the NHL does not record the type of shot immediately preceding a deflection/tip-in. It is therefore likely that more of the shots displayed in these figures result in a goal than the corresponding shots displayed in the goal heatmaps (Figures 12 and 13). This again supports the notion that a more complex xG metric is required for hockey compared to football since taking a shot from increased distance is not directly proportional to a decrease in the probability of a goal on the same play/next touch of the puck - instead also depending on the original shot type.

5.6 Conclusion of the Analysis

From the data analysis, four key areas have been identified to calculate team statistics for to construct the feature vectors of the models: penalties, puck possession, shot type and goals scored.

The number of penalties a team concedes likely increases the number of goals conceded. This was shown in the team strength analysis where the second most goals scored are by the team on the powerplay (i.e. have the player advantage). A statistic must therefore be considered that captures not only the number of penalties for a team, but also how well they manage the penalty (e.g. how often do they concede a goal while shorthanded).

Puck possession be determined from the number of shots taken: a team taking more shots compared to the opposing team should in theory have more possession. More shots implies an offensively stronger side, therefore pushing the opposition back into their defensive zone.

This statistic could further be refined to only consider shots that have a chance of scoring (i.e. no missed shots). This captures how offensively clinical the team is, since one side could have greater puck possession but few shots that challenge the goalie (therefore decreasing the expected number of goals scored).

The number of shots of a certain type can influence the expected number of goals a team scores, as highlighted in the conversion rate of shot types (Table 18). In its most simple version, a team taking more tip-ins / creating the most deflections should score more due to the higher chance of these shots resulting in a goal.

Finally, the number of goals scored so far by a team should provide some indication on how many goals the team will score in the future (i.e. the

next game being predicted). This helps to capture a team's form: a team that consistently scores well should continue to score well in the near future. This can be applied to game prediction since a team with good form should outcompete a team with a comparatively weaker form.

The difference in number of games played per season means that these statistics can only be computed on a season-long basis. As was shown in Figure 6, varying the number of games played also varies the total number of goals scored that season. As such, data cannot be accumulated between seasons due to the potential of introducing some form of skew; a team may score more goals than the previous season but this is only due to playing more games and does not indicate an improvement in team performance.

For this reason, statistics for the teams of the game currently being predicted will be calculated on a season-cumulative basis; only games played by the team that season, up until the game being predicted, will be considered in the statistic.

6 Machine Learning Models

The use of machine learning techniques was chosen for this project due to their ability to identify patterns in large datasets that can be used to classify individual examples. The problem to be solved is binary classification of games into the result *'win'* or *'loss'*, therefore such techniques are well suited to help solve this.

Eight different machine learning models were trained and compared in order to maximise the accuracy of predictions. These can be split into traditional methods and ensemble methods:

Traditional Methods

- Decision Tree
- Naïve Bayes
- Support Vector Machine (SVM)
- Neural Network
- Stochastic Gradient Descent (SGD)

Ensemble Methods

- AdaBoost
- Gradient Boosting
- XGBoost

The inclusion of ensemble methods was chosen due to their common ability to improve the accuracy of initially weak classifiers. Previous work by

Weissbock et al. [1] has shown that pre-existing classifiers achieve an accuracy around the mid 50% range. Although a weak classifier is generally considered one with an accuracy of $50\% + \epsilon$ for some small positive constant ϵ, these models may still be weak enough to benefit from boosting methods.

Each of the models were implemented with the use of the scikit-learn library [19] with the exception of XGBoost [33] which is available as its own library.

Hyperparameter tuning was achieved with the use of the grid search method. This involves iterating through every possible combination of parameter values defined within the search space. Although this is computationally complex, it allowed for the optimal combination of parameters to be discovered to maximise predictive accuracy.

Tuning was implemented with the scikit-learn `GridSearchCV` class. By defining the parameter grid as a dictionary, where keys are the hyperparameters provided by the respective scikit-learn model implementation, the default 5-fold cross-validation grid search is applied to the training set. The `fit()` method is then called to train the models using the optimal combination of values founds. This is defined as the combination which results in the greatest mean accuracy of the 5-fold cross validation on the training set; the optimal combination can be accessed through the `bestparams` attribute.

Input data for some models, such as SVM, is required to be normalised since this is assumed by their scikit-learn implementation. This is common due to objective functions being defined for features that follow a Gaussian distribution, so that no feature dominates the learned weights.

For simplicity, all data is therefore normalised using the scikit-learn `Standard Scaler` class; this centres and normalises the data to a new set with unit variance on a feature-by-feature basis.

The performance of each model will mainly be evaluated based on its accuracy metric since this is used by most other work, simplifying the comparison to our models' performance. This metric is also the most intuitive to use since the aim of the project is to maximise the proportion of correctly predicted game results.

The accuracy metric is defined by the following formula:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives.

6.1 Decision Tree

Decision trees attempt to break down the given dataset into smaller subsets based on splitting rules for the features. This is repeated, in training, until each leaf consists of a subset where every example has the same ground truth.

Predictions are then made for new examples by applying these rules to determine which subset the data is most similar to. The ground truth of this subset is the prediction.

This model is implemented in scikit-learn with the `sklearn.tree.DecisionTreeClassifier` [34] with default parameters. This results in all tree nodes being expanded until all leaves contain only data with the same ground truth; the time required to predict new games may therefore increase due to a larger depth of the tree, however this should increase accuracy.

Instead of the information gain approach used to measure the quality of the data split for a given rule, the Gini impurity is utilised. This metric indicates the likelihood of new data being misclassified using the split's class distribution. This has been chosen due to the label only being binary, therefore randomness in the labels should have a larger influence on accuracy.

Due to the high dimensionality of the data and the potential of introducing noise depending on feature vector implementation (discussed in the *Game Prediction* section), it is unlikely that this model will provide the greatest accuracy. However, this is still included for its simplicity and can be considered a good baseline.

6.2 Naïve Bayes

Naïve Bayes is a probabilistic model which uses conditional independence assumptions of the features to determine the probability of a label given the set of features.

This is implemented using the scikit-learn `sklearn.naivebayes.GaussianNB` [35] class.

Due to the simplicity of this model, training should not take too long, making the model appropriate for the relatively large dataset. In addition, Naïve Bayes is known to often work well for smaller datasets in a range of applications. This will be useful for when the original dataset is restricted to a single season (*Single-Season Data Approach*).

However, since each feature in an example is a game statistic, it is highly unlikely that the independence assumption is valid. This may therefore significantly decrease the accuracy of the model.

6.3 Support Vector Machine

Support Vector Machines attempt to determine the hyperplane that separates two classes of data while maximising the distance to the closest point (support vector) in each class.

An SVM is implemented using the scikit-learn `sklearn.svm.SVC` [36]. The tuned hyperparameters are explained below.

As the problem is one of binary classification, an SVM seems well-suited since we want a model that can clearly separate the data into two classes (win/loss). Also, the maximisation of the decision boundary should improve the generalisation for test data. This is especially useful for the prediction of an entire season's results in advance.

SVMs are also effective for handling high-dimensional data, such as that used in this project, due to the use of the kernel trick.

6.3.1 Hyperparameters

Hyperparameter	Values
C	[0.1; 100]
γ	[0.0001; 1]
Kernel	[linear, rbf]

Table 19: Tuned Hyperparameters for SVM

C is the regularisation parameter: the strength of the squared L2 penalty is inversely proportional to this.

γ is the kernel coefficient for the RBF kernel. This value is inversely proportional to the standard deviation (σ) of the kernel.

Two kernels have been chosen: the linear and RBF (Radial Basis Function) kernels to model both linear and non-linear data respectively. This should ensure that the model returns the highest accuracy, regardless of the trend the data follows.

The RBF kernel is defined for two points X_1, X_2 by the following formula:

$$RBF(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2})$$

6.4 Neural Network

A Multi Layer Perceptron (MLP) is a type of Neural Network formed from multiple layers of artificial neurons. Input data is fed to the first layer where pre-activation values are calculated by the weighted sum of the input. Each neuron then applies a non-linear activation function to this sum before passing the output to the next layer. The network may consist of multiple hidden layers before the label is determined at the output layer. The weights used in the pre-activation sums are updated through backpropagation, which uses a cost function during the training phase.

The MLP is implemented with the scikit-learn `sklearn.neuralnetwork.MLPCLassifier` [37] and hyperparameters are explained below.

Due to the use of the non-linear activation function, Neural Networks are able to approximate a wide variety of functions. This is therefore appropriate for the project as the size of the dataset should allow for the optimal function that best defines the datapoints to be found.

6.4.1 Hyperparameters

Hyperparameter	**Values**
α	[0.001; 1]
Activation	[identity, logistic, tanh, relu]
Batch Size	[10; 200]
Max Iterations	[10; 400]

Table 20: Tuned Hyperparameters for NN

The value of α represents the strength of the L2 regulariser, with a higher value indicating increased generalisation (as weights are encouraged to approach zero).

Multiple common activation functions have been selected to introduce non-linearity into the features (note that 'logistic' refers to the sigmoid function). The identity function has also been included in case the data already consists of linear relationships. Although it is not assumed that this will provide the most accurate results, it is included for completeness.

Batch size is the size of the minibatches used to train the network. Increasing the size of the minibatch should decrease time to convergence, however increases the chance of overfitting. A good balance is therefore required.

The max iterations determines the maximum number of epochs allowed. Increasing the number of epochs increases the tuning of the network weights to optimal values through backpropagation and increases the chances of convergence. Too high a number however can lead to overfitting as training data is 'memorised' - models perform badly on new data.

6.5 Stochastic Gradient Descent

Gradient Descent uses the negative gradient of the cost function to optimise the weights to minimise cost. This process is linear in the number of examples. Since the training data used in this model contains a large number of games, this can become very expensive.

Stochastic Gradient Descent (SGD) overcomes this by only selecting one random example to compute the negative gradient. The cost of computing this is now independent of the size of the dataset.

This model is implemented in scikit-learn by the `sklearn.linearmodel.SGDClassifier` [38] with hyperparameters explained below.

6.5.1 Hyperparameters

Hyperparameter	Values
α	[0.0001; 1]
Loss	[log, hinge, huber, squared hinge, perceptron, squared error]
Penalty	[None, L1, L2, elasticnet]
Max Iterations	[100; 1000]

Table 21: Tuned Hyperparameters for SGD

Loss refers to the loss function to minimise in order to maximise the accuracy of the model. A wide range of losses with varying penalisation on incorrect classifications have been considered in order to maximise the search space of the grid search and determine the best combination of parameters. Note that the 'perceptron' loss is equivalent to the linear loss function utilised by the perceptron algorithm.

Penalty is the regulariser to apply to the weights in an attempt to generalise the model. Both L1 and L2 regularisation are considered in order to evaluate whether sparsity may improve accuracy by ignoring unimportant features. Elasticnet is a combination of L1 and L2 which attempts to fuse the benefits of both methods.

6.6 AdaBoost

AdaBoost (Adaptive Boosting) attempts to manipulate the training set to improve the accuracy of an initially weak classifier.

The original classifier is first trained on the original training set, with equal weights for each example. These sample weights are then updated, with those classified incorrectly becoming more important. This process is repeated with subsequent training sets weighted to emphasise the more difficult examples. The final model is therefore a weighted combination of each of these trained classifiers.

This ensemble technique is implemented in scikit-learn with `sklearn.ensemble.AdaBoostClassifier` [39] and its hyperparameters are explained below.

6.6.1 Hyperparameters

Hyperparameter	Values
Estimators	[10; 1000]
Learning Rate	[0.0001; 1]

Table 22: Tuned Hyperparameters for AdaBoost

Estimators is the maximum number of classifiers that can be sequentially trained before the final model is output. If 100% accuracy is achieved, then further classifiers are not added. Increasing this value should therefore increase the accuracy of the final model but may also increase the model complexity if convergence is slow.

The learning rate is the weight applied to each weak classifier during training, with a greater weight resulting in a greater contribution by the classifier to the final model.

6.7 Gradient Boosting

Similar to AdaBoost, Gradient Boosting sequentially builds the final model with each pass of the training set. This is achieved by fitting a regression tree on the negative gradient of the loss function.

Initially, a weak regression tree is constructed from the training data (which outputs continuous values rather than a class label). This allows for further trees to be combined by summing these outputs. Gradient descent is utilised to ensure that these added trees help to minimise the loss. This attempts to correct the final prediction of the model.

Gradient Boosting tends to be more robust to overfitting, therefore it is expected that this model will report higher accuracies than AdaBoost.

This algorithm is implemented in scikit-learn with the `sklearn.ensemble.GradientBoostingClassifier` [40]. Its hyperparameters are described below.

6.7.1 Hyperparameters

Hyperparameter	**Values**
Estimators	[10; 1000]
Learning Rate	[0.0001; 1]
Max Depth	[1; 10]
Min Samples Split	[0.1; 1.0]
Min Samples Leaf	[0.1; 0.5]

Table 23: Tuned Hyperparameters for Gradient Boost

Max Depth is the maximum depth that each regression tree can reach. Increasing this will increase the likelihood of reaching leaves containing data of only one class, however also increases the complexity of the model (and chances of overfitting).

The Minimum number of Samples per Split is tuned to change how many remaining examples are required to form another node. Increasing this value helps to restrict the depth of the tree and therefore model complexity.

The Minimum number of Samples per Leaf is a similar parameter, however considers the number of samples in both branches. A new node is created if and only if both contain at least this number. This again restricts the depth and also helps to maintain a more balanced tree.

6.8 XGBoost

XGBoost (Extreme Gradient Boosting) is a further implementation of Gradient Boosting. This method incorporates regularisation in an attempt to reduce the chances of overfitting, improving accuracy. Regression trees are fit with the use of the squared error to determine additional trees to add to minimise the loss.

Regularisation is usually applied with the use of the L1 norm. Since this encourages weights to equal zero, this restricts the number of leaf nodes possible per tree (as features with zero weight cannot be split), and hence tree depth. As a result, the complexity of the model is restricted.

This method can be implemented with its own library [33], separate to scikit-learn. The algorithm is contained within the `xgboost.XGBClassifier`. The hyperparameters are explained below.

6.8.1 Hyperparameters

Hyperparameter	**Values**
Estimators	[10; 1000]
Learning Rate	[0.001; 1]
Max Depth	[1; 10]
γ	[0; 2]
Column Samples per Tree	[0.1; 1]

Table 24: Tuned Hyperparameters for XGBoost

The value of γ represents the minimum reduction in loss required for a split to be made. Increasing this value therefore only allows a split when a significant improvement is made to the final model. This helps to generalise and reduce the chance of overfitting.

The Column Samples per Tree determines the proportion of features that are randomly sampled at each node during training. Reducing this fraction should in turn reduce the chances of overfitting, as only a subset of features per example are considered. This however will reach a limit where the model will begin to decrease in accuracy as the inter-feature relationships cannot be captured.

7 Game Prediction

With all required data collected and hyperparameters identified, the models could be trained and tested using a variety of feature vectors.

Two main approaches were explored for the use of the collected data:

- Multi-Season Data Approach (all collected game data is used to train/test models).
- Single-Season Data Approach (only game data from the 2021/22 season is used to train/test models).

The second approach was not initially considered, however became apparent as a possibility when analysing the accuracies of the multi-season data models. This led to an investigation into whether multi-season data should in fact be used to predict future games. This is further explained in the subsequent subsections.

Using the multi-season data approach, the goal of improving model accuracy was achieved with the use of feature engineering, with four feature vectors being suggested (each an iterative change on the previous).

The accuracy of each are compared to determine the optimal predictor that is also robust to new data, along with an explanation of the intuition behind the addition/removal of each feature to/from the vector.

7.1 Multi-Season Data Approach

The initial approach was to use all data collected from the nine seasons to train and test the models. All data from 2013/14 - 2020/21 was isolated as the training set, with the 2021/22 season the test set to calculate model accuracy. This accuracy score is measured as the percentage of game results predicted correctly as win/loss.

One of the reasons to use 8 seasons worth of training data is to analyse the impact of increasing the size of the training set on game prediction. This is discussed further in the *Basic Feature Vector* section. The method was inspired by the work of Gu et al. [4] whose increase in collected data between 2007/08 - 2016/17 led to models of a much higher accuracy than previously published.

The use of such a large training set is to try and identify general underlying trends in the game of hockey that can result in good game prediction. Multi-season data should reduce the impact of good form in a single season or a team experiencing a run of 'lucky' results that do not reflect their statistics of the time.

If no general trends are found, and hence poor accuracy in the test set, then this also leads to another conclusion: multi-season data cannot be used to accurately predict the outcome of games in the current season. An investigation into whether this is true is explored in the *Single-Season Data Approach*.

As this was the first method for the use of the collected data, this is when all feature engineering took place - with the proposal of four separate feature vectors:

- Basic Feature Vector
- Advanced Feature Vector
- Reduced Noise Feature Vector
- Ten-game Feature Vector

Each of these intended to try and result in improved predictive performance from the previous model. The intuition behind each vector is explained in their own subsequent subsection.

The creation of each new feature vector was implemented with the use of pandas Dataframes [17]. The collected data was read from the csv files into separate Dataframes. From this, the required statistics to be used as features could be calculated and inserted into a new Dataframe on a game-by-game basis covering the entire 9 seasons. This new dataset was then split into the training and test sets by including the Game ID in the data - this attribute was removed from the final feature vectors before being fed into the models as this feature does not impact game outcome, instead only helping to identify games.

7.1.1 Basic Feature Vector (2 Vectors per Game)

The first feature vector constructed was based upon the initial work by Weissbock et al. [1], with the calculation of features similar to those included in their vector. The choice of creating such a similar feature vector is to provide a benchmark accuracy score, from which, further feature engineering attempts to improve on. The accuracy recorded in the original work cannot directly be used as the benchmark due to the difference in size and time of their collected data. Our much larger dataset should reduce the risk of overfitting which may skew their stated accuracies from data only being collected over a three month period. Also, this data was from the 2012/13 season, nine years previous to the most recent season included in our dataset. It is possible that a shift in the style of play occurred over this time, meaning that statistics included as features now should be assigned a different weighting on more recent data.

The use of this similar feature vector also helps to explore whether purely increasing the size of the dataset alone (from three months to nine years) helps to improve model accuracy. If this is a significant increase, then it would suggest that future applications of such models in the sport may continue to see some increase in accuracy. This would also hint at the presence of constant underlying trends in hockey statistics that could be prioritised in

analysis by coaches in order to improve gameplay and win percentage.

Another implication of this would be to provide some insight into how much of the increase in model accuracy published by Gu et al. [4] was due to the increase in size of their dataset (also a period of nine seasons) or, more likely, a result of their testing method or expert system approach.

Stat	**Explanation**
Team	Long name of the team
Location	'Home' or 'Away'
Goals For	Number of goals the team has scored in the season so far.
Goals Against	Number of goals the team has conceded in the season so far.
Goal Difference	Goals For minus Goals Against so far this season.
Powerplay Percentage	Percentage of powerplays the team scored on so far this season.
Penalty Kill Percentage	Percentage of penalty kills the team prevented the opposition scoring on so far this season.
Shot Percentage	Percentage of shots scoring so far this season.
Save Percentage	Percentage of shots saved by the team's goalie so far this season.
Winning Streak	Number of consecutive games won this season before the current game.
Fenwick Close Percentage	Percentage of the number of unblocked shots taken by the team out of all shots in games so far this season while in a close game.
PDO	A measure of luck, calculated by a team's season Shot Percentage + Save Percentage.
5-5 F/A	The proportion of goals for to goals against while the team is at even strength.

Table 25: Basic Feature Vector

Table 25 lists all 12 identified features along with a brief explanation of the statistic.

Each statistic was calculated for a game using a season-cumulative method.

Each game corresponds to two feature vectors, one for each participating team. For each team, all games that the team has competed in so far that season are used to calculate the statistics. This approach was chosen in an attempt to capture the season-long form of a team. Some seasons, a team may generally underperform compared to their average due to a variety of factors (e.g. club money, coaching staff, player transfers etc.). A model that uses statistics carried over from previous seasons may inaccurately predict many games involving said team as it would appear that the team is still strong compared to others this season, due to the temporal lag in the change of statistical values.

A consequence of this approach is that games played earlier in the season may be harder to predict due to all teams sharing similar valued statistics, with these being initialised to average values. This is similar to the cold-start problem [41] that recommender systems may suffer from - the result of a lack of data relating to new items. Statistics of the first games played in the season are shared by both teams, therefore predictive accuracy should be bound to 50% (i.e. random guessing) with a slow improvement from this as the spread of statistical values increase over the course of the season.

To mitigate this issue, the *Team* name is included in the feature vector. This is implemented as a one-hot encoding of the 32 teams (i.e. the team name feature actually consists of 32 features all set to the value 0, except the relevant team which is set to value 1). As this does not change between seasons, teams that consistently perform well throughout previous seasons are likely to be weighted in favour of winning future games. The first games of the season are therefore predicted based on general historical performances only.

This method also applies to games at the end of the season. Since all teams play 82 games per season, statistics tend to regress back to the norm over time. For most teams that are neither dominating nor underperforming

consistently in games, this means that there is again less of a spread of data to distinguish between likely winners. Although there is still a greater spread compared to the start of the season, the use of team name may be relied upon again to aid the prediction based on past performance.

The *Location* feature identifies whether the given team is playing home or away. In many sports, the home team is often regarded as having a slight advantage due to more experience playing in their stadium/arena and the psychological benefit of a large home support throughout the game.

This advantage may be inflated in the NHL due to the large distances travelled by some teams for interconference games (e.g. the distance between the Seattle Kraken and Boston Bruins is 2489 miles / 4005km as the crow flies). The significant travel time required may impact player performance as the team could arrive to the game tired and slightly cramped from the journey. Location is implemented as a one-hot encoding of home/away in an attempt to capture these potential influences.

Identified in the *Data Analysis*, the number of goals a team has scored so far may be a good indication of future goals scored and therefore probability of winning a game. This is captured in the *Goals For*, *Goals Against* and *Goal Difference* features.

The original analysis only concludes the necessity for the Goals For statistic. This is a good measure of a team's offensive capability, with a higher count indicating a stronger performance. To increase the chances of winning, a team must also have a strong defence to prevent the opposition from scoring; just scoring a lot of goals does not guarantee a win if the opposition also scores highly. This shows the importance of including the Goals Against count.

Each of these two statistics provide an insight into the offensive and defensive abilities of the team separately. These can be combined into the *Goal*

Difference stat which provides an insight into team performance generalised across all opposition faced. A team with a higher and positive goal difference is more likely to score more and therefore win the game than one with a lower and negative difference.

It could be argued that only the Goal Difference is required as a feature. This however could cause an issue when teams with similar differences play each other. In this case, the team who has previously scored more goals is likely to continue to score more goals in the next game and therefore win, showing the importance of recording Goals For.

Since scoring goals is the only datapoint that directly corresponds to winning a game, it was decided to include all three of these features to attempt to maximise predictive accuracy.

Team strength was another factor to consider from the analysis. This is captured in the *Powerplay Percentage* and *Penalty Kill Percentage* features. It would have been possible to capture the penalty events through the Penalties In Minutes statistic, which is a cumulative count of the number of penalty minutes accrued by a team. A team with more penalty minutes is shorthanded more often therefore is more likely to be scored on by the opposition (from Figure 7 in the analysis).

This however is not a completely robust conclusion as a team may be defensively strong on the penalty kill; although the team may have poor discipline, they can mitigate this by often preventing the opposition from scoring during the penalty kill. A statistic must therefore be used to indicate a team's performance on the powerplay/penalty kill instead of solely identifying the number of these occurrences.

Since both sides are highly likely to concede a number of penalties during a game, the team with the higher proportion of goals scored on powerplays should therefore have more good scoring opportunities. This team's probability of scoring can further be increased if the opposition concedes on a

higher proportion of penalty kills. The Powerplay Percentage and Penalty Kill Percentage both capture these proportions.

As previously mentioned, the common football xG metric [2] does not consider the number of shots taken by a team but rather the shot quality, with more shots of a higher quality increasing the number of expected goals. Instead of using a location-based implementation, we have chosen to initially consider the percentage of shots that lead to a goal with the *Shot Percentage* statistic. A higher Shot Percentage should correlate to a greater proportion of higher quality shots; assuming an equal skill level amongst opposition goalies, shots should only score if they are of a good enough quality, or the play immediately preceding the shot is strategically well executed to reduce the goalie's saving ability.

Similar to the goal-based statistics, we must not only consider the offensive aspect of the team but also the defensive capability - specifically that of the team's goalie. A goalie that concedes a higher proportion of shots will require the team to have an increased Shot Percentage to match the opposition. The *Save Percentage* feature captures the goalie's ability at shot stopping.

Although the season-long cumulative statistics implicitly record a team's season form, this is further refined to the most recently played games by the *Winning Streak* feature. Team's can often experience a run of games with poor performance, before returning to their average win percentage. Explanations for such dry spells can range from individual player injury, to facing a run of fixtures against typically stronger opponents. By reducing the scope of the recorded form (and therefore increasing its specificity) to a count of the number of consecutive wins immediately before the next game, this may indicate whether a team is going through an exceptionally over/under performing run of games. This should therefore improve the prediction of

individual games at different points of the season.

Another conclusion from the data analysis was the importance of representing puck possession, since the team with more possession should in theory also have more scoring opportunities. A method of measuring puck possession can therefore be derived from the proportion of shots a team has out of all shots taken during a game.

The Fenwick Percentage is one such statistic which refines the shot idea to only consider shots that travel through the defence (therefore have some chance of scoring and can be considered as more influential to game outcome). This is therefore equivalent to all shots except those recorded as *'Blocked'* by the NHL.

The following equation shows this calculation using the NHL stats API [20] shot outcome terms (where A and B are the teams competing in this game):

$$Fenwick(A) = \frac{Goals(A)+Saved(A)+Missed(A)}{\Sigma AllShots(A)+\Sigma AllShots(B)}$$

This statistic can further be refined to the Fenwick Close Percentage, which uses the same formula but only considers shots taken in a 'close game situation'. This is defined as a one goal score difference or less in the first two periods, and a tied game in the third period. The motivation for this is to provide a more accurate measure of puck possession depending on the context of the game.

Typically, a team trailing by multiple goals will attempt, and often succeed, at gaining more puck possession in their desire to make a comeback and score more goals. In response, the winning team will sit back in defence and try to hold their ground. These events could skew the Fenwick Percentage statistic in favour of the losing team; this may result in models predicting the opposite result for each team. Hence, the *Fenwick Close Percentage* has been chosen as the measure of puck possession.

Mentioned in the *Introduction*, 'luck' is often regarded as having a larger influence in hockey games than compared to other sports. This is a relatively difficult feature to define quantitatively to be able to incorporate the idea into models.

One method is to use the *PDO* statistic. The sum of a team's shot and save percentage should, over time, regress to 100% since most teams in the League have players of a similar skill level. As such, there is little range amongst team shot and save percentages. Throughout the season however, there will be some fluctuation in these statistics. A team may happen to begin scoring from more of their shots than usual, increasing their shot percentage. This could be a result of increased deflections or lucky bounces due to imperfections in the ice surface etc. Likewise, a team's goalie may begin to save more shots than on average. Explanations for this could be that the goalie just happens to be in the right position to face every shot, or manages to make a blind save by spreading their body as wide as possible.

Both of these events will result in an increase in shot and save percentages, thus increasing PDO above 100%. The team can be said to be experiencing more luck in this period of time, with results happening to fall their way. This is slightly different to the concept of 'form' which is instead often attributed to an increase in the skill of the team as a whole and can be seen in an improvement of most statistics. Luck however is only observed in a change in statistics relating to goals scored. Similar to form, this could have some indication (although probably less certain) that the team will win games in the near future, despite having relatively poor stats that do not directly correlate with goals scored.

Whilst the importance of measuring statistics relating to the performance of teams gaining/serving penalties has been noted due to the player advantage/disadvantage, it is also necessary to consider team performance at even

strength. This is because penalties are rare events, compared to total game time, meaning most of the game is played with five-on-five skaters. Consequentially, most goals are therefore scored when teams are at even strength (Figure 7).

This performance can be captured by the ratio of a team's goals scored to goals against while at even strength; this is the *5-5 F/A*. Values greater than 1 indicate a team that is strong both offensively and defensively for the majority of the game, with even higher values suggesting that this strength is more equal in both aspects. Values less than 1 indicate both a weak offense and defence, therefore these teams are less likely to win games.

7.1.2 Basic Feature Vector (1 Vector per Game)

There are potential drawbacks with the two feature vector per game implementation, as mentioned in the literature review of the initial work from Weissbock et al. [1].

The chances of a team winning a game are not only dependent on the statistics of said team but also the statistics of the opposition. The winner is likely to be the side with generally better statistics in comparison. This is important in determining the outcome of a contest between two similarly weak or dominant teams; teams must not both be predicted a win or loss, which may be possible when considering the teams in isolation (i.e. two feature vectors).

This comparison of statistics can be achieved by combining the two feature vectors into one, with the home team statistics placed first, followed by the away team's. The models then attempt to predict the outcome of the home team only (win/loss); this therefore also implicitly predicts the outcome of the away team as the opposite class (due to draws not being possible in the NHL regular season).

The models were trained and tested with the use of both Basic Feature

Vector implementations. For each model, the accuracy of predictions on the test set were recorded; these are shown Tables 26 (for the untuned models) and 27 (using tuned hyperparameters).

Untuned Model Results:

Model	1 Vector Accuracy	2 Vector Accuracy	Diff
DT	51.6%	50.3%	+1.3%
NB	55.8%	53.1%	+2.7%
SVM	55.9%	55.2%	+0.7%
NN	50.5%	52.4%	-1.9%
SGD	53.7%	56.7%	-3%
AdaBoost	58.5%	57.1%	+1.4%
Gradient Boosting	**61.4%**	56.8%	+4.6%
XGBoost	55.9%	53.3%	+2.6%

Table 26: Untuned Model Accuracy of Basic Feature Vector Implementation

Tuned Model Results:

Model	1 Vector Accuracy	2 Vector Accuracy	Diff
SVM	57.5%	58.7%	-1.2%
NN	61.4%	56.1%	+5.0%
SGD	**62.7%**	59.1%	+3.6%
AdaBoost	62.3%	59.0%	+3.3%
Gradient Boosting	60.1%	59.4%	+0.7%
XGBoost	61.3%	58.7%	+2.6%

Table 27: Tuned Model Accuracy of Basic Feature Vector Implementation

The maximum tuned accuracy of the Gradient Boosting model with the 2 feature vector per game implementation of 59.4% is only 0.02% higher than

that produced in the work of Weissbock et al. [1]. Due to the similarity of the feature vectors, this shows that increasing the size of the dataset has a negligible impact on game prediction, hence it is unlikely that either work's implementation of the models are largely overfitting on a specific feature. This also suggests that both implementations are classifying results based on similar trends, confirming the existence of metrics that generally determine game outcome.

The 1 vector per game implementation of the feature vector is significantly more accurate, with the SGD model achieving 62.7% (an increase of 3.6%). Neural Networks saw the largest increase in accuracy of 5% (equal to 65 more correct predictions).

This supports the reasons proposed for the combination of the two vectors in order to utilise statistics of both teams in the prediction. From this success, the one feature vector implementation will be used for further feature engineering.

7.1.3 Advanced Feature Vector

With the success of the one feature vector implementation of the Basic Feature Vector, more statistics were sought after to include as additional features in order to improve model accuracy.

Three more features (Table 28) were identified as providing potentially beneficial data for the models. The addition of these were based on the findings in the data analysis which emphasised the importance of puck possession statistics and shot types. These statistics were calculated using the same season-cumulative method in the previous vector.

Stat	Explanation
Corsi Percentage	Percentage of shots taken by the team out of all shots taken so far this season
Deflected Shots	Number of deflected shots so far this season
Tip-In Shots	Number of tip-in shots so far this season

Table 28: Advanced Feature Vector (additional only)

Another method of interpreting puck possession is the *Corsi Percentage*. Similar to the Fenwick Percentage, this measures puck possession based on the proportion of shots taken by each team in a game, with the team with the higher proportion also likely having the greater puck possession.

Whereas Fenwick Percentage does not include blocked shots, Corsi Percentage considers all shots taken by each team (A and B):

$$Corsi(A) = \frac{Shots\ Team\ A}{Shots\ Team\ A + Shots\ Team\ B}$$

The intuition behind Fenwick Percentage is to only consider shots that could pose a threat to the goalie by actually passing through the entire defence. Blocked shots however signify the potential danger if the defence were to fail, since these shots would otherwise most likely end up as saved (since this is the highest shot outcome in Figure 8). Corsi Percentage therefore considers the attacking potential of a team rather than just past evidence of a team's shot threat on goal.

Since we are predicting future games, we should be incorporating this idea of potential threat into the model as it is not possible to be certain of how the opposition's defence will perform on the night.

Blocked shots also offer the opportunity of a second shot on goal from a different angle. These shots may also have an improved chance of scoring

since the goalie has just reacted to the first shot, therefore has less chance to setup to face the second in a changed location.

Data analysis also highlighted how different shot types have varying conversion rates associated with them. Table 18 showed how tip-ins and deflections have significantly higher conversion rates of 17.8% and 17.7% respectively. A team managing to take more of these shots should subsequently score more goals. A count of the number of shots of each of these two types (taken so far in a given season) were therefore added to the feature vector as the *Deflected Shots* and *Tip-In Shots* features.

The addition of these features formed the Advanced Feature Vector. Tables 29 and 30 display the accuracy of the models using this vector. A comparison to the best accuracy of the Basic Feature Vector models is also provided to analyse whether the performance has been improved.

Untuned Model Results:

Model	Original Accuracy	Added Features Accuracy	Diff
DT	51.6%	52.1%	+0.5%
NB	55.8%	56.1%	+0.3%
SVM	55.9%	56.9%	+1.0%
NN	50.5%	51.7%	+1.2%
SGD	53.7%	54.2%	+0.5%
AdaBoost	58.5%	56.8%	-1.7%
Gradient Boosting	**61.4%**	60.4%	-1.0%
XGBoost	55.9%	54.3%	-1.6%

Table 29: Comparison of Basic and Advanced Feature Vector Implementation on Untuned Model Accuracy

Tuned Model Results:

Model	Original Accuracy	Added Features Accuracy	Diff
SVM	57.5%	58.5%	+1.0%
NN	61.4%	61.6%	+0.2%
SGD	**62.7%**	61.9%	-0.8%
AdaBoost	62.3%	61.2%	-1.1%
Gradient Boosting	60.1%	61.5%	+1.4%
XGBoost	61.3%	60.7%	-0.6%

Table 30: Comparison of Basic and Advanced Feature Vector Implementation on Tuned Model Accuracy

Although the maximum accuracy of the Advanced Feature Vector (61.9% from the tuned SGD model) did not improve on the 62.7% accuracy from the Basic Feature Vector, the majority of models did however show a slight increase. This suggests that this implementation will be more robust to changes in the data; since the test set is independent to the training set, an average increase in accuracy implies that the models, which use varied methods, are able to classify new data more accurately.

For this reason, this implementation is considered as a successful improvement to the feature vector and will be used for additional feature engineering.

These results do however imply that the shot count of the high conversion rate shot types is not as influential in game outcome as originally suggested. This may be due to the difficulty in an attacking team to force tip-ins and deflections to occur, compared to other shots that the player can decide to take for themselves.

7.1.4 Reduced Noise Feature Vector

With the incorporation of the two feature vectors into one, there is now the potential of the introduction of unnecessary noise into the dataset.

The *Game Location* feature actually consists of a one-hot encoding of length 2 to represent home/away. This information however is now implicit in the one feature vector per game implementation; the first set of statistics are for the home team, the second set for the away team, and we are now only predicting the outcome of the home team. The location information is therefore redundant and can be removed without influencing the prediction (as these values are the same for every vector).

The *Team* name feature is also a one-hot encoding of length 32 to represent every team in the league. This is an almost three times increase in the dimensionality of the feature vector, vastly increasing the complexity of the models. The outcome of a game also does not depend on the team name but rather the statistics of the competing teams. There is therefore a chance that the models may be overfitting on team name, rather than placing more weight on the statistical features. This is most noticeable if team performance greatly differs between seasons; a team may suffer from an unusually poor season, however is predicted their usual wins due to their team name alone.

Although the team name was originally introduced to the feature vector to help with the prediction of early-season games, the significant decrease in dimensionality of the vector may be of greater benefit than high accuracy in a minority of the total games predicted.

These features were removed from the Advanced Feature Vector, with the new Reduced Noise Feature Vector then being passed to the models to be retrained and tested. The accuracies achieved are shown in Tables 31 and 32. Similar to previous tables, a comparison to the accuracy achieved by the Advanced Feature Vector is provided to determine whether the new vector exhibits an improvement in predictions.

Untuned Model Results:

Model	Advanced Features Accuracy	Reduced Features Accuracy	Diff
DT	52.1%	53.0%	+0.9%
NB	56.1%	**61.7%**	+5.6%
SVM	56.9%	60.6%	+3.7%
NN	51.7%	54.5%	+2.8%
SGD	54.2%	58.5%	+4.3%
AdaBoost	56.8%	58.2%	+1.4%
Gradient Boosting	60.4%	59.2%	-1.2%
XGBoost	54.3%	53.4%	-0.9%

Table 31: Comparison of Advanced and Reduced Noise Feature Vector Implementation on Untuned Model Accuracy

Tuned Model Results:

Model	Advanced Features Accuracy	Reduced Features Accuracy	Diff
SVM	58.5%	60.1%	+1.6%
NN	61.6%	61.7%	+0.1%
SGD	**61.9%**	61.7%	-0.2%
AdaBoost	61.2%	60.7%	-0.5%
Gradient Boosting	61.5%	58.7%	-2.8%
XGBoost	60.7%	60.7%	±0%

Table 32: Comparison of Advanced and Reduced Noise Feature Vector Implementation on Tuned Model Accuracy

Similar to the step from the Basic to Advanced Feature Vector, there is again no increase in maximum accuracy between the Advanced and Reduce Noise vectors, with the best reaching 61.7% with Naïve Bayes. Although this is a second decrease in maximum accuracy, this implementation again generally produced an increase in the accuracies of most models, most significantly in those untuned.

Naïve Bayes saw the largest increase of 5.6%. This again implies that these models may be more robust to data. The majority of tuned mod-

els now also consistently predict over 60% of games accurately. This value was considered a milestone at the beginning of the project, surpassing this determining the project's success.

7.1.5 Collection of Statistics from Previous 10 Games Only

The previous feature vectors all utilised the method of season-long cumulative statistics: all previous games played by the team in the given season are used to calculate each statistic. This provides a general insight into the performance of the team that season. However, due to the number of games played by each team in a season, most of these statistics tend to regress towards their averages over time. This can make it difficult to distinguish the performance of teams compared to others and therefore decreases the confidence in predictions.

The range in statistical values can be increased by reducing the amount of time considered prior to games for calculating statistics. Using only the past 10-games a team participated in before the game in question, the statistics are refined to capture the current form of the team - something not possible in season-long statistics.

This method also means that each game is given equal importance in the statistics. Using the season-long approach, the influence on statistical values of games played later on in the season weakens due to the increase in total number of games played. By consistently only considering ten games, each game contributes a tenth in terms of importance to the statistics (each game in this method is weighted equally).

Note that games occurring before the 10th game of the season played by a team use cumulative statistics from the start of the season since not enough games have been played so far.

The features for this implementation had to be recalculated from the original collected data. The data necessary to calculate the statistics was stored

based on a queue of size ten; when new data is collected, the oldest game data is removed - this reduces the complexity of having to recalculate rolling ten-game stats.

The models were then provided with the new feature vector in order to be retrained. The accuracies from the test set are shown in Tables 33 and 34. A comparison to the previous Reduced Noise Feature Vector is also provided to determine whether this feature vector improves predictive accuracy.

Untuned Model Results:

Model	Reduced Features Accuracy	Ten Game Data Accuracy	Diff
DT	53.0%	52.0%	-1.0%
NB	**61.7%**	59.8%	-1.9%
SVM	60.6%	53.7%	-6.9%
NN	54.5%	55.6%	+1.1%
SGD	58.5%	55.6%	-2.9%
AdaBoost	58.2%	59.1%	+0.9%
Gradient Boosting	59.2%	59.8%	+0.6%
XGBoost	53.4%	54.3%	+0.9%

Table 33: Comparison of Advanced and Reduced Noise Feature Vector Implementation on Untuned Model Accuracy

Tuned Model Results:

Model	Reduced Features Accuracy	Ten Game Data Accuracy	Diff
SVM	60.1%	59.6%	-0.5%
NN	**61.7%**	61.4%	-0.3%
SGD	61.7%	61.1%	-0.6%
AdaBoost	60.7%	61.3%	+0.6%
Gradient Boosting	58.7%	59.1%	+0.4%
XGBoost	60.7%	59.2%	-1.5%

Table 34: Comparison of Advanced and Reduced Noise Feature Vector Implementation on Tuned Model Accuracy

A further decrease in maximum accuracy has resulted from the Ten Game Feature Vector implementation, with the Neural Network only achieving 61.4% accuracy. However, unlike the other iterations, more models have seen a significant decrease in accuracy, such as -6.9% for the SVM. These levels have been deemed to be unacceptable therefore this iteration is considered as a failure and provides no useful gain to the predictive capability of the models.

The decrease suggests that the current form of the team does not greatly influence the outcome of the proceeding game - each game could be considered as independent in this sense. It may still be possible however to refine the method for capturing form. It is likely that more recent games will still have a slightly greater influence on game outcome than earlier games. The data used in past games could therefore be weighted when calculating the statistics prior to the next game, with earlier games contributing less to this value than the latest game. This could be achieved through some form of geometric progression that tends to 0 as the number of games played since that being weighted increases.

7.1.6 Overall Comparison Between Models

The following Tables (35 and 36) summarise the results obtained in this section. The maximum accuracy achieved was 62.7% with SGD using the 1 feature vector per game implementation of the Basic Feature Vector.

The Reduced Noise Feature Vector provided the most consistent increase, with the minimum accuracy being 58.7% with Gradient Boosting. It is this feature vector that is considered to be the most successful as it should be the most robust to new data when applied to future seasons. This maximises the future potential of the work produced in this project.

Untuned Model Results:

Model	1 Vector	2 Vector	Advanced	Reduced	Ten Game
DT	51.6%	50.3%	52.1%	53.0%	52.0%
NB	55.8%	53.1%	56.1%	**61.7%**	59.8%
SVM	55.9%	55.2%	56.9%	60.6%	53.7%
NN	50.5%	52.4%	51.7%	54.5%	55.6%
SGD	53.7%	56.7%	54.2%	58.5%	55.6%
AdaBoost	58.5%	57.1%	56.8%	58.2%	59.1%
Gradient Boosting	61.4%	56.8%	60.4%	59.2%	59.8%
XGBoost	55.9%	53.3%	54.3%	53.4%	54.3%

Table 35: Comparison of Feature Vector Implementation on all Untuned Model Accuracy

Tuned Model Results:

Model	1 Vector	2 Vector	Advanced	Reduced	Ten Game
SVM	57.5%	58.7%	58.5%	60.1%	59.6%
NN	61.4%	56.1%	61.6%	61.7%	61.4%
SGD	**62.7%**	59.1%	61.9%	61.7%	61.1%
AdaBoost	62.3%	59.0%	61.2%	60.7%	61.3%
Gradient Boosting	60.1%	59.4%	61.5%	58.7%	59.1%
XGBoost	61.3%	58.7%	60.7%	60.7%	59.2%

Table 36: Comparison of Feature Vector Implementation on all Tuned Model Accuracy

7.2 Single-Season Data Approach

Since feature engineering alone was not providing the desired increase in accuracy of game prediction, it was decided to modify how the collected data was being used.

Although the work produced by Gu et al. [4] also collected 9 seasons of

game data, only a random subset of this was sampled for use in training and testing (which equated to a season's worth of games).

A similar approach was now taken for this project with the dataset being limited to the 2021/22 season (the most recent collected data). The training and test set were selected using an 80/20 split: a random 80% of games played (1049) used as training data and the remaining 20% (263) as test data. This was achieved using the scikit-learn `traintestsplit()` method with a `randomstate` of 42.

The main reason behind this change is that team performance may differ greatly between seasons, therefore patterns in past statistics may not be applicable to future games. This is more notable in hockey than other sports such as football due to the high number of transfers in the off-season which can result in a significantly changed team the following season.

A greatly changed team will likely work well with a different style of play compared to previous seasons. This different style may skew the team's statistics compared to the common trends, however the team may still perform well; the models however may predict this outlier as consistently losing.

The idea of a differing playing style is also possible due to a change of coaching staff in the off-season. Often, coaches have different perspectives on how best to play the game to maximise win percentages; some may play more defensively, focusing on powerplays or fast breakouts when puck possession is won. As a result, a team's statistics can again be significantly different between seasons, meaning that past data is less relevant to predicting future results. Instead, only the current season's games may provide any useful information.

The NHL differs from football in that it implements the idea of 'trades' [42]. A player from one team can simply be traded for one from another. This reduces the cost of transfers which have now reached tens of millions in the top leagues of football. As a result, trades can be performed easily

by most clubs frequently, meaning that many teams will likely reflect this change in multiseason data.

To analyse the impact on predictive accuracy of reducing the dataset to one season, all feature vector implementations have been used to retrain and test the models with the new dataset. The best accuracy of these is then compared to the maximum accuracy from the multi-season dataset for each model in Tables 37 and 38.

Untuned Model Results:

Model	**Best Multi-season**	**Ten Game**	**Reduced**	**Advanced**	**Max Diff**
DT	53.0%	50.6%	53.2%	52.5%	+0.2%
NB	61.7%	60.1%	55.1%	65.4%	+3.7%
SVM	60.6%	64.3%	63.1%	63.9%	+3.7%
NN	55.6%	55.5%	64.3%	56.3%	+8.7%
SGD	58.5%	62.0%	65.4%	58.2%	+6.9%
AdaBoost	59.1%	61.6%	64.3%	63.5%	+5.2%
Gradient Boosting	61.4%	62.0%	59.7%	62.0%	+0.6%
XGBoost	55.9%	56.3%	61.6%	62.7%	+6.8%

Table 37: Comparison of Feature Vector Implementation on Untuned Model Accuracy Using 21/22 Season Data

Tuned Model Results:

Model	Best Multi-season	Ten Game	Reduced	Advanced	Max Diff
SVM	60.1%	63.5%	67.3%	64.3%	+7.2%
NN	61.7%	67.3%	**69.2%**	66.5%	+7.5%
SGD	62.7%	66.9%	67.3%	66.2%	+4.6%
AdaBoost	62.3%	64.3%	64.3%	65.8%	+3.5%
Gradient Boosting	61.5%	63.1%	66.5%	66.5%	+5.0%
XGBoost	61.3%	64.3%	65.0%	64.3%	+3.7%

Table 38: Comparison of Feature Vector Implementation on Tuned Model Accuracy Using 21/22 Season Data

Reducing the dataset to only consider the 2021/22 season resulted in a significant increase in prediction accuracy to a maximum of 69.2% with Neural Networks using the Reduced Noise Feature Vector. This is an increase of 6.5% on the best multi-season model (SGD).

The magnitude of this increase strongly suggests that only the current season provides useful information as to the probability of a team winning a game - likely due to the change in teams in the off-season leading to a large change in performance. As a result, it is concluded that seasons can be considered as independent with past performance having little impact on future games. This is an important consideration if applying the models to future seasons beyond those recorded in the dataset.

8 2021/22 Season Final Division Standings Prediction

From the success of the models using the 2021/22 season data only, we now attempt to predict the points and Division standings of all teams competing that season.

The final standings and points of each team are separated into Divisions instead of the League as a whole for easier analysis of the model's performance. Due to there being fewer teams in each Division (8), a small change in the number of points predicted has a lower chance of altering the team's final Division ranking compared to their final League ranking.

Predicting only Division rankings also benefits coaching staff since these rankings determine qualification for the Stanley Cup Playoffs. More importance is placed on the Playoffs than the League in the NHL, therefore it is usually the ambition of most coaches to achieve qualification. Hence, it is useful for staff to be able to predict their team's final standings to determine whether the current performance is good enough to reach this series.

Predicting the number of points also provides coaching staff with further data on their performance. The number of points a team is behind their rivals is a good indicator of how much improvement is required and can be used as a method of measuring their intensity of wins.

Two prediction methods were explored: predicting the whole of the 2021/22 season, and only predicting the final 20% of the 2021/22 season. This second method was used in an attempt to improve the accuracy of the final standings and is discussed further in its own section.

It should be noted that the models do not consider Overtime losses, from which a team would gain a single point instead of 0. As such, when collecting the ground truth values of the number of points actually accrued by each team in the 2021/22 season, this data was manipulated to remove the gains

of such losses.

8.1 Simulation of the Whole 2021/22 Season

Predicting a season in advance is likely to be the most common practical use of the models instead of individually predicting games. Predicting single games is often useful for deciding a team to place a bet on, however this is an easier decision for someone to make due to the recency of past games played by each team; it is easier for someone to make a judgement on a team's chances of winning based on their previous results.

Higher odds bets are often offered for predicting end-of-season rankings, especially predicting eventual champions. This is where a model with good season-long prediction is of benefit, since little is known about current team form to predict a team's performance after 1312 games of the season.

Using the results from the previous training of the models on the single-season dataset (Table 38), the model with the highest accuracy score was chosen to be used to predict the 2021/22 season games. In this instance, the AdaBoost model using the one vector per game implementation of the Basic Feature Vector was chosen as this provided the most consistent results in testing.

Predicting the season's games produced the confusion matrix in Figure 16. This matrix shows the count of each category a prediction can fall under when compared to the ground truth of each example: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

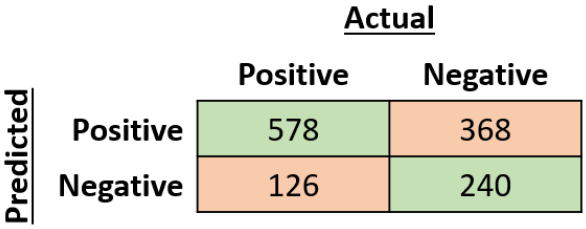

Figure 16: Whole Season Prediction Confusion Matrix

It can be seen from the matrix that a significantly larger proportion of games were predicted a win (946) than loss (366). Since the model predicts the outcome of the home team, this would suggest that the home team has a 72% chance of winning. The common held belief that the home team has an advantage is therefore supported by this trend.

From the confusion matrix, two metrics can be calculated: precision and recall. Precision is the proportion of positive values (wins) correctly predicted out of all positive predictions. Recall is the proportion of positive values correctly predicted out of all positive ground truth values. The formulae for these metrics are shown below:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

The model provides a precision of 61.1% and a recall of 82.1%. The higher recall value tells us that the model is managing to identify the majority of actual wins, however the lower precision indicates the model is also overpredicting this number (a significant portion of losses are predicted as wins).

Tables 39 - 42 show the predicted standings of each Division at the conclusion of the 2021/22 regular season. For each team, the predicted number

of points gained is displayed. Also, the difference in standing position and number of points accrued in the ground truth are shown. Where the correct standing is predicted, the team name and difference is highlighted in green. If the predicted points is also correct, this value is also highlighted.

Atlantic Division

Team	Predicted Pts	Diff
Toronto Maple Leafs	128	+1 (+20 pts)
Florida Panthers	126	-1 (+10 pts)
Boston Bruins	116	±0 (+14 pts)
Tampa Bay Lightning	116	±0 (+14 pts)
Ottawa Senators	44	±0 (-22 pts)
Detroit Red Wings	34	±0 (-30 pts)
Montréal Canadiens	28	+1 (-16 pts)
Buffalo Sabres	24	-1 (-40 pts)

Table 39: Predicted 21/22 Atlantic Division Standings (Whole Season)

Metropolitan Division

Team	Predicted Pts	Diff
Carolina Hurricanes	120	±0 (+12 pts)
Pittsburgh Penguins	120	+1 (+28 pts)
Washington Capitals	116	+1 (+28 pts)
New York Rangers	98	-2 (-6 pts)
Columbus Blue Jackets	68	+1 (-6 pts)
New Jersey Devils	58	+1 (+4 pts)
New York Islanders	52	-2 (-22 pts)
Philadelphia Flyers	40	±0 (-10 pts)

Table 40: Predicted 21/22 Metropolitan Division Standings (Whole Season)

Central Division

Team	Predicted Pts	Diff
St. Louis Blues	120	+2 (+22 pts)
Colorado Avalanche	118	-1 (+6 pts)
Minnesota Wild	106	-1 (±0 pts)
Nashville Predators	102	+1 (+12 pts)
Winnipeg Jets	82	+1 (+4 pts)
Dallas Stars	70	-2 (-22 pts)
Chicago Blackhawks	22	±0 (-34 pts)
Arizona Coyotes	14	±0 (-36 pts)

Table 41: Predicted 21/22 Central Division Standings (Whole Season)

Pacific Division

Team	Predicted Pts	Diff
Edmonton Oilers	130	+1 (+32 pts)
Calgary Flames	120	-1 (+20 pts)
Los Angeles Kings	114	±0 (+26 pts)
Vegas Golden Knights	102	±0 (+16 pts)
Anaheim Ducks	76	+2 (+14 pts)
San Jose Sharks	64	±0 (±0 pts)
Vancouver Canucks	56	-2 (-24 pts)
Seattle Kraken	40	±0 (-14 pts)

Table 42: Predicted 21/22 Pacific Division Standings (Whole Season)

The standings of each division are fairly accurate with 12 teams being predicted the correct position and 14 only being inaccurate by one place. The greatest inaccuracy in standings is by 2 positions for the remaining 6 teams.

The predicted points however are considerably less accurate, with each Division following a similar pattern: teams at the extremes of the table are

significantly over/underpredicted points and teams in the middle are generally more accurate in points. This is clearly shown in Table 41 which shows the predicted standings of the Central Division. The St. Louis Blues are overpredicted by 22 points (11 wins) and the Arizona Coyotes underpredicted by 36 points (13 losses). The mid-table teams are comparatively more accurate in point predictions with the Winnipeg Jets only being overpredicted 4 points (2 wins).

This suggests that the performances of the more dominant and weaker teams are being inflated compared to the ground truth (and hence a potential source of the low precision suffered by this model). A benefit of this analysis however is that it is clear the models can separate the performances of a stronger team from those consistently playing poorly (which is the basis to forming a model that can predict a winning team). This separation must now be refined to reduce the False Positives/Negatives, which will also improve precision.

As the model chosen for the simulation uses the one feature vector per game implementation of the Basic Feature Vector, the exaggeration of the top and bottom team's performance may be a result of the model overfitting on the Team Name. If a weighting places too much importance on this feature, then it is possible that the St. Louis Blues were mainly predicted wins just because they are the team that are playing, without much focus on the statistics. As such, the Reduced Noise Vector may be of better use in future simulations.

Since only the extremes of the table are consistently exaggerated, it may be possible to achieve improved points accuracies by reducing the number of games predicted. This can be done by beginning the simulation at a later point in the 2021/22 season.

8.2 Simulation of Final 20% of the 2021/22 Season

In an attempt to improve the accuracy of the final Division standings and points, the simulation of the season is restricted to only the final 20% of games played in the 2021/22 season (263 games).

For this method, the Neural Network model using the Reduced Noise feature vector was chosen due to its consistently high accuracy during testing the single season dataset (Table 38). This model was trained on the first 80% of games played that season (1049 games). By then predicting games from the end of the same season, we expect to see an increase in the accuracy of the predicted Division standings and points - similar to the increase in accuracy seen as we moved from the multi-season dataset to the single season approach.

		Actual	
		Positive	Negative
Predicted	Positive	91	76
	Negative	56	40

Figure 17: Final 20% of Season Prediction Confusion Matrix

Figure 17 displays the confusion matrix for this simulation. The predictions resulted in a precision of 54.5% and a recall of 61.9%. Both of these values are lower than those from the season-long simulation. That said, the predicted rankings and points are noticeably more accurate than previous, with the Pacific Division showing 100% accuracy in predicted standings (Table 46).

This discrepancy is most likely due to the overpredicting of wins. The reduced recall indicates that fewer of the actual wins are being identified. However, it can be seen in the confusion matrix that the number of False Positives (76) is relatively high. This means that the loss of points from missing wins are being counteracted by an increase in the number of incorrectly predicted wins. Since the final standings are only based on number

of points, and not whether individual games between teams are predicted correctly, this helps to close the gap from lost points and results in standings closer to the ground truth.

This highlights how the performance of simulations should not solely be used to evaluate the accuracy of models. A model may be fairly inaccurate, however since only the points earned by teams are used to predict rankings, it is possible that (by chance) enough false positives are predicted to cancel out the false negatives.

Tables 43 - 46 show the predicted standings of each Division at the conclusion of the 2021/22 regular season, predicted once 80% of the season is complete. Again, teams are highlighted using the same method as the previous whole season simulation.

Atlantic Division

Team	Predicted Pts	Diff
Florida Panthers	106	±0 (-10 pts)
Toronto Maple Leafs	104	±0 (-4 pts)
Boston Bruins	100	±0 (-2 pts)
Tampa Bay Lightning	92	±0 (-10 pts)
Detroit Red Wings	68	+1 (+4 pts)
Buffalo Sabres	64	+1 (±0 pts)
Ottawa Senators	56	-2 (-10 pts)
Montreal Canadiens	50	±0 (+6 pts)

Table 43: Predicted 21/22 Atlantic Division Standings (Final 20%)

Metropolitan Division

Team	Predicted Pts	Diff
Carolina Hurricanes	102	±0 (-6 pts)
New York Rangers	100	±0 (-4 pts)
Pittsburgh Penguins	92	±0 (±0 pts)
Washington Capitals	90	±0 (+2 pts)
Columbus Blue Jackets	82	+1 (+8 pts)
New Jersey Devils	72	+1 (+18 pts)
New York Islanders	68	-2 (-6 pts)
Philadelphia Flyers	54	±0 (+4 pts)

Table 44: Predicted 21/22 Metropolitan Division Standings (Final 20%)

Central Division

Team	Predicted Pts	Diff
Colorado Avalanche	108	±0 (-4 pts)
Minnesota Wild	98	±0 (-8 pts)
Nashville Predators	96	+2 (+6 pts)
Winnipeg Jets	88	+2 (+10 pts)
Dallas Stars	86	-1 (-6 pts)
St. Louis Blues	84	-3 (-14 pts)
Chicago Blackhawks	64	±0 (+8 pts)
Arizona Coyotes	60	±0 (+10 pts)

Table 45: Predicted 21/22 Central Division Standings (Final 20%)

Pacific Division

Team	Predicted Pts	Diff
Calgary Flames	98	±0 (-2 pts)
Edmonton Oilers	94	±0 (-4 pts)
Los Angeles Kings	84	±0 (-4 pts)
Vegas Golden Knights	84	±0 (-2 pts)
Vancouver Canucks	80	±0 (±0 pts)
San Jose Sharks	76	±0 (+12 pts)
Anaheim Ducks	70	±0 (+8 pts)
Seattle Kraken	54	±0 (±0 pts)

Table 46: Predicted 21/22 Pacific Division Standings (Final 20%)

The models achieved 100% accuracy in prediction of the standings in the Pacific Division with both the Vancouver Canucks and Seattle Kraken also being predicted the correct number of points (Table 46). The points prediction of the top 4 teams also only differed by an underprediction of 1 or 2 wins. This suggests that this Division was more predictable compared to others that season. This is an interesting observation due to the decrease in range of predicted points (44 points) in contrast to the other divisions (maximum of 56 points in the Atlantic Division - Table 43).

An initial implication would be that the Pacific Division was highly contested, with all teams winning/losing a similar number of games (excluding Seattle Kraken with only 54 points). This would therefore suggest that it is more difficult to accurately predict the winner of intradivision games. Thus, it is likely that more interdivison games were predicted correctly, possibly as a result of the teams in this division being generally weaker compared to others therefore easier to classify (hence the lowest number of points in all divisions which secured first place - 98 points).

In general, the teams at the extremes of the tables also saw an improved accuracy in points predicted. For example, the whole season simulation over-

predicted the Calgary Flames points total by 12.2% of the predicted games (Table 42). The reduced simulation only underpredicted the points by 6.25% of predicted games (Table 46).

The doubling in this accuracy (in proportion to the number of games predicted) shows that the model performs significantly better in this simulation. This is likely due to the availability of previous game data from the current season at the beginning of the simulation. This therefore does not suffer from the previously discussed 'cold start problem', which is likely where most of the over/underpredictions occur.

These simulations have shown that it is possible to predict the final standings of the NHL Division a whole season in advance with an accuracy of ± 1 position for most teams. If an accuracy in the number of points gained is required, then this simulation must be restricted to a set period of time during the season that is currently in progress. In turn, this also increases the accuracy of overall standings.

9 Investigating the Impact of the COVID-19 Pandemic on Game Prediction

As mentioned in the data analysis, the 2019/20 and 2020/21 seasons were impacted by the COVID-19 pandemic, resulting in fewer games than usual being played (1082 and 868 respectively). Figure 6 showed that this, in turn, significantly reduced the number of goals scored during these seasons.

In an attempt to maximise the number of games played, the 2020/21 season also implemented a temporary division realignment in order to mitigate the impact of Canadian travel restrictions [7] (see Table 47). As a result, teams played intradivisional games only for a 56 game season each. The top four teams of each Division qualified for the Stanley Cup Playoffs, which was played in its usual format.

West	**Central**	**North**	**East**
Ducks	Hurricanes	Flames	Bruins
Coyotes	Blackhawks	Oilers	Sabres
Avalanche	Jackets	Canadiens	Devils
Kings	Stars	Senators	Islanders
Wild	Red Wings	Maple Leafs	Rangers
Sharks	Panthers	Canucks	Flyers
Blues	Predators	Jets	Penguins
Golden Knights	Lightning		Capitals

Table 47: 2020/21 Temporary Division Realignment

The change of fixtures meant there was more repetition in the teams faced than usual. This could potentially skew the statistics for the season if a team consistently played poorly as this would increase the offensive stats of other teams at a faster rate than in a usual season. This could lead to models trained on previous season's data being poor at predicting this season. Alternatively, this could negatively impact the models' predictions of future

seasons, if trained on this season.

There is also the potential impact of the disease on individual players. If tested positive for COVID-19, players were required to self-isolate in accordance with the NHL's self-isolation protocol [43]. This meant that teams would be without some players at several points of the season. Coaches would then have to form a new strategy to ice a team which may be less suited to their usual style of play. As a result, it is likely that team performance is subject to fluctuate more during this season, hence being less predictable.

In addition to the individual players' period of self isolation, there is also the possibility of an extended period of reduced player performance from the effects of 'Long Covid'. Research into the potential effects of this was conducted by Kuitunen et al. [44] with data from Finnish leagues. The conclusion from this research was that not only could players suffer from the effects of COVID-19 beyond the self-isolation period, but that also there was a high risk from asymptomatic players unknowingly spreading the disease to the opposition. This therefore increases the impact on player performance not only on the length of time of recovery, but also on the rate of spread of the impact between teams.

It is therefore necessary to investigate the impact of these seasons on predictive ability. For example, it may be suboptimal to include this data in the overall dataset and should not be considered for predicting future seasons.

This investigation was conducted by applying two methods:

- Excluding the COVID-19 seasons from the dataset
- Only considering the COVID-19 seasons as the dataset

These methods are further discussed in the following sections.

9.1 Considering All Except COVID-19 Seasons

The first approach was to remove the 2019/20 and 2020/21 seasons from the dataset and re-evaluate the accuracy of the models. An increase in the average accuracy would suggest that the new dataset is more easily separable; as this new dataset only considers the 'normal' seasons, this would in turn imply that the COVID-19 seasons do not follow this shared trend and can be considered as outliers to the data.

The models were retrained on the new dataset of 8774 games with the use of the Reduced Noise feature vector implementation. Similar to the method implemented in the 2021/22 season simulation, an 80/20 train/test split was utilised (resulting in a training set of 7019 games and a test set of 1755 games).

The accuracies of the models trained on this new dataset are shown in Tables 48 and 49, along with a comparison to the maximum accuracies of the original multi-season dataset.

Untuned Model Results:

Model	Accuracy Including COVID-19 Seasons	Accuracy Excluding COVID-19 Seasons	Diff
DT	53.0%	52.9%	-0.1%
NB	61.7%	61.7%	±0%
SVM	60.6%	61.2%	+0.6%
NN	54.5%	55.3%	+0.8%
SGD	58.5%	60.1%	+1.6%
AdaBoost	58.2%	58.2%	±0%
Gradient Boosting	59.2%	59.5%	+0.3%
XGBoost	53.4%	53.4%	±0%

Table 48: Comparison of Untuned Models Trained on Data Excluding COVID-19 Seasons

Tuned Model Results:

Model	Accuracy Including COVID-19 Seasons	Accuracy Excluding COVID-19 Seasons	Diff
SVM	60.1%	60.1%	±0%
NN	61.7%	62.4%	+0.7%
SGD	61.7%	62.7%	+1.0%
AdaBoost	60.7%	60.7%	±0%
Gradient Boosting	58.7%	61.5%	+2.8%
XGBoost	60.7%	60.7%	±0%

Table 49: Comparison of Tuned Models Trained on Data Excluding COVID-19 Seasons

All models reported an increase in predictive accuracy except the Decision Tree (decrease of 0.1%). The greatest improvement in accuracy was from the Gradient Boosting model (+2.8%) and the maximum accuracy for the Reduced Noise feature vector increased from 61.7% to 62.7% with Stochastic Gradient Descent.

This increase in accuracy, without the inclusion of the COVID-19 seasons, suggests that these seasons may be outliers to the general trends in the 'normal' seasons. The inclusion of these seasons in the original dataset may have skewed the data, resulting in the models having greater difficulty to assign appropriate weights to the features and separate the games into the two classes.

The improved predictions therefore suggest that the 2019/20 and 2020/21 seasons should not be included in multi-season datasets due to the pandemic influencing game outcome to a significant degree when compared to other seasons.

9.2 Considering COVID-19 Seasons Only

The second part of this investigation was on restricting the dataset to only include data from the 2019/20 and 2020/21 seasons. This was to measure the predictability of the games played during these seasons.

If the potential impacts of the pandemic actually influenced game outcome, then it is likely that these games would be harder to predict. This is because it is not possible to include the extraneous effects as features in the model; the models therefore have no knowledge of this data and cannot consider its impacts in the classification of results.

The relative predictability of these seasons is measured by comparing model accuracy on the COVID-19 seasons only to the model accuracy of the 'normal' seasons only. If the COVID-19 model accuracy is significantly lower,

then it can be concluded that the pandemic did have an impact on game outcome. These accuracies are displayed in Tables 50 and 51.

Untuned Model Results:

Model	**Best Accuracy Excluding COVID-19 Seasons**	**Accuracy of COVID-19 Seasons Only**	**Diff**
DT	52.9%	55.8%	+2.9%
NB	61.7%	60.5%	-1.2%
SVM	61.2%	60.3%	-0.9%
NN	55.3%	55.4%	+0.1%
SGD	60.1%	58.9%	-1.2%
AdaBoost	58.2%	58.4%	+0.2%
Gradient Boosting	59.5%	57.2%	-2.3%
XGBoost	53.4%	55.6%	+2.2%

Table 50: Comparison of Untuned Models Trained on COVID-19 Seasons Data Only

Tuned Model Results:

Model	Excluding COVID-19 Seasons	Accuracy COVID-19 Seasons Only	Diff
SVM	60.1%	58.9%	-1.2%
NN	62.4%	59.6%	-2.8%
SGD	62.7%	55.6%	-7.1%
AdaBoost	60.7%	58.6%	-2.1%
Gradient Boosting	61.5%	59.3%	-2.2%
XGBoost	60.7%	61.0%	+0.3%

Table 51: Comparison of Tuned Models Trained on COVID-19 Seasons Data Only

When considering only the COVID-19 seasons' data, there is a significant decrease in the predictive accuracy, especially in the tuned models. All of these experienced a decrease (except a 0.3% increase with XGBoost), with the greatest being suffered by the Stochastic Gradient Descent model (-7.1%).

This large decrease suggests that games played during these seasons are less predictable than those in 'normal' seasons. This is especially supported by the fact that the SGD model was the most accurate for normal season data, however saw the greatest decrease on COVID-19 season data.

We can therefore conclude that games played during the COVID-19 seasons are more unpredictable than previous games. Since the collected data, feature vectors and models were unchanged, it is likely that the decrease in accuracy is due to some extraneous influences as a result of the pandemic. These two seasons can therefore be considered as outliers to the general trends observed in the normal seasons and should not be considered in multi-season datasets.

10 Future Work

This section suggests possible extensions to the work completed in this project and potential methods for achieving this. The Stanley Cup Playoffs Prediction and Betting Model are further explanations of the extensions originally considered for this project in the Specification [27], however were replaced with the investigation on the impact of the COVID-19 pandemic.

10.1 Stanley Cup Playoff Prediction

With the success of the prediction of the final 20% of the 2021/22 season, it is possible to easily implement a similar method to predict the outcome of the Stanley Cup Playoffs.

The dataset can be restricted to include only games played that season by teams that have qualified for the playoffs - this should halve the size of the dataset. The First Round ties will be known in advance as these depend on the ground truth standings of the Divisions. Each tie can therefore be predicted by the model, with the winner progressing to the next round. This prediction is repeated for each of the four rounds, the final prediction determining the overall winner.

There is however an issue with this suggested implementation. Playoff rounds consist of best-of-seven series. Based on the regular season data alone, the model would predict the same winner for each of these seven potential games as no new data is being added to the dataset (assuming the Playoffs are being predicted in advance). This means that individual playoff games cannot be predicted, instead only the winner of the round as a whole.

Being unable to predict the individual games means that it is not possible to explore whether playing the same team multiple times may have an impact on game prediction. For example, teams may either dominate by quickly understanding how to outplay other teams, or ties may be tightly contested all the way to Game 7. If this specific question is being considered,

it may therefore be more useful to predict past Playoff series, with the dataset being updated after each game in the same way as the regular season. This however restricts Playoff prediction to a round-by-round basis as inter-round data depends on the ground truth of which teams win the previous series.

A playoff round is played in both team's arena in the following locations per game: home-home-away-away-home-away-home, where the home team is the higher ranked side (giving them an overall home advantage). If using a feature vector that considers the game location, the home team should therefore be considered as the higher ranked team due to the extra home game in the round.

10.2 Location-Based Expected Goals Metric

An expected goals metric more similar to that often implemented in football [2] is theoretically possible (however likely to have low accuracy as previously discussed). Potential methods of achieving this are now suggested.

Further analysis of the heatmaps produced to visualise locations on the ice where goals are scored indicate a significant clustering directly in front of the goal. This region can be captured within the shape of a trapezium: the short side being the goal mouth with the diagonals extending towards the two offensive zone faceoff circles.

Figure 18 shows this trapezium overlayed on the heatmap of all goals scored between 2013/14 - 2021/22. This same trapezium is then overlayed onto the hockey rink diagram in Figure 19 to better visualise the locations where most goals are scored from.

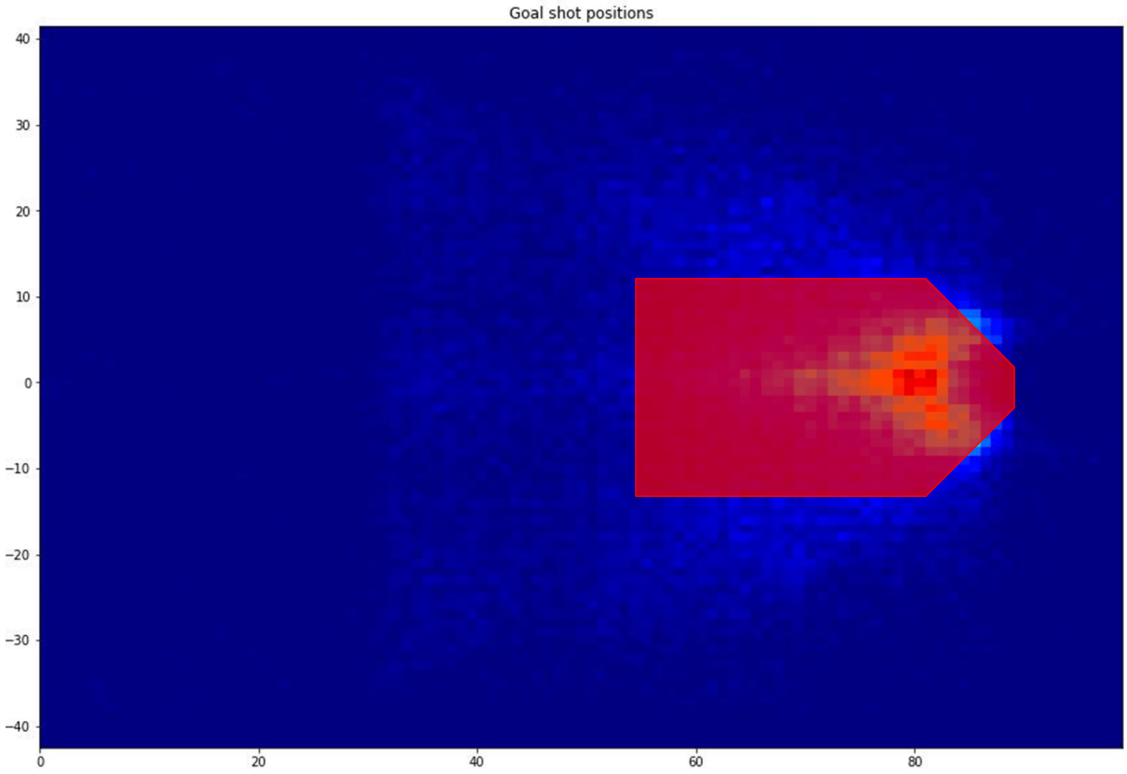

Figure 18: Optimal Region to Score Goals

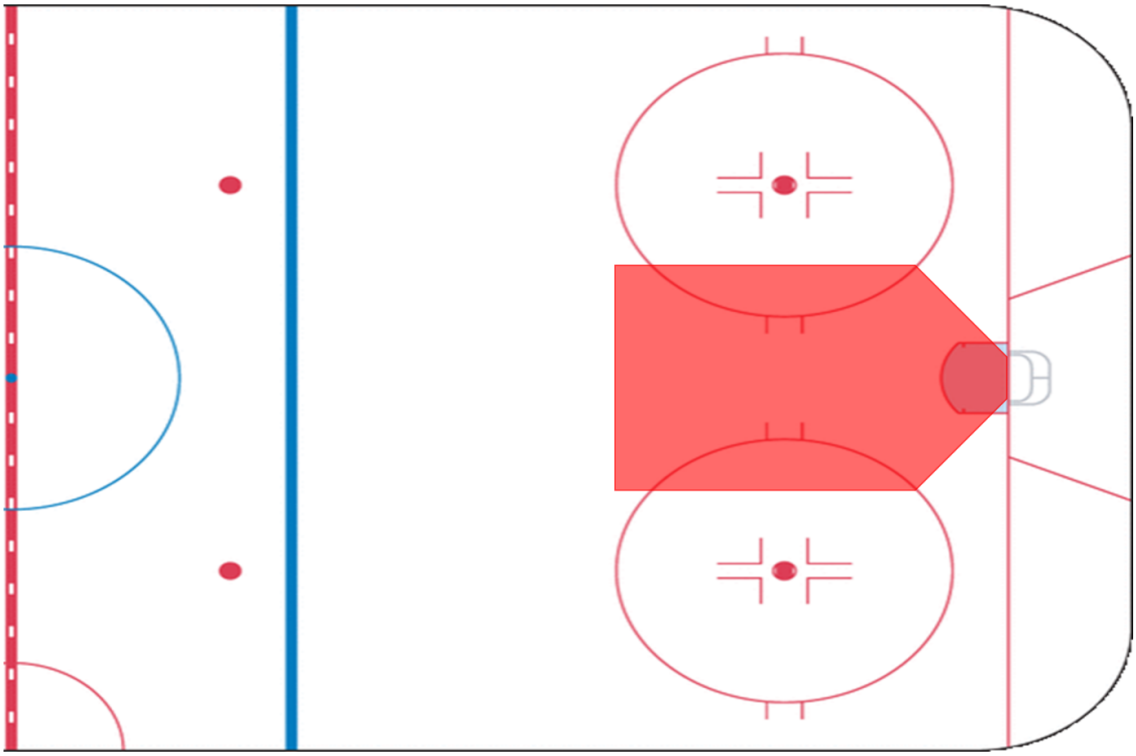

Figure 19: Optimal Region to Score Goals with Rink Superimposed

Consider a regression model that calculates the probability of scoring a goal from a given shot. The coordinate of the shot could be considered as a feature, with coordinates located within this trapezium at either end of the rink (associated with the attacking team) being weighted to increase the probability of scoring. Those occurring outside of this region should be weighted to result in a lower goal probability.

Care must be taken to ensure that the models do not overfit on this shot location. Other factors may also greatly influence shot outcome. For example, different shot types have different conversion rates therefore can be taken at varying locations with varying success. In addition, as shown in the original analysis (Figure 8), 49.4% of the shots that are on target will likely be saved by the goalie regardless of where the shot was taken. Shot location

cannot therefore be the sole feature to consider for an expected goals metric.

Within the identified trapezium region, there is further variation on the number of goals scored at different locations. It can be seen that significantly more goals are scored directly in front of the goal at close distance than those further away at decreased angles to the goal mouth. The location feature can therefore be extended into greater detail by individually including the distance and angle of the shot to the centre of the goal.

Figure 20 demonstrates how these values are obtained from the x and y coordinates of a shot location on the rink. Each value has an associated range depending on where in the goal the shot is aimed at. It is unlikely that the shot will always be aimed at the centre of the goal as this is where the goalie is usually positioned to cover most of the goal mouth. As a result, the y coordinate of each shot varies by ±3 units due to the goal mouth covering 6 units in this direction.

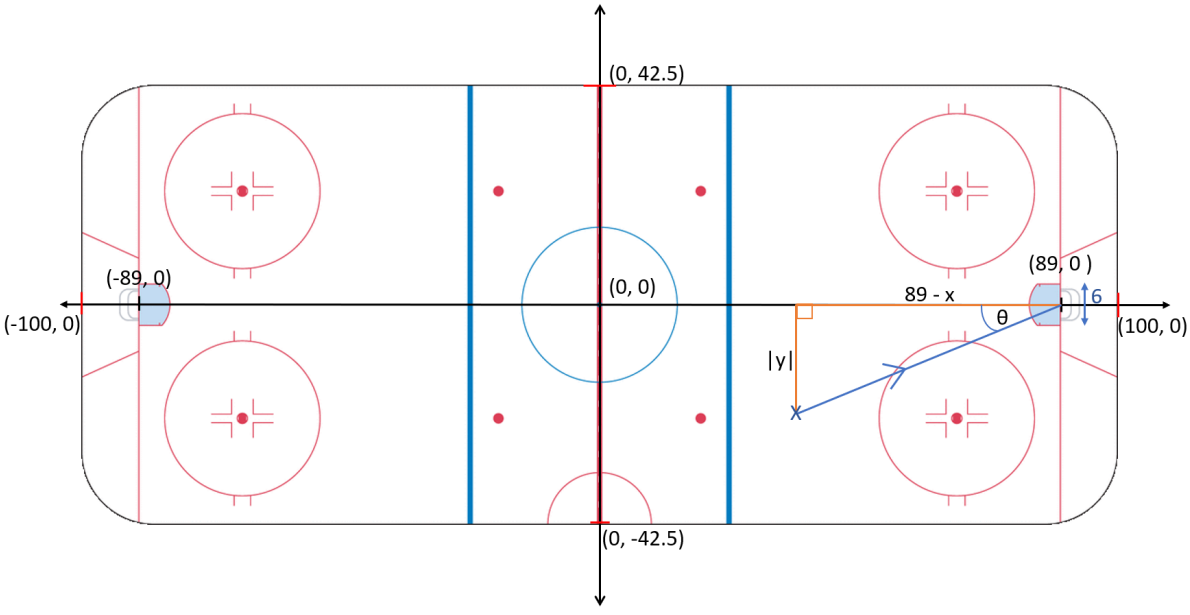

Figure 20: Diagram of Shot Positioning

From the diagram in Figure 20, the following equations can be derived to determine the range of distance and angle values for a given shot location:

$$\sqrt{(|y|-3)^2+(89-x)^2} \leq d \leq \sqrt{(|y|+3)^2+(89-x)^2}$$

$$tan^{-1}(\tfrac{|y|-3}{89-x}) \leq \theta \leq tan^{-1}(\tfrac{|y|+3}{89-x})$$

Where d is the distance the puck travels to the goal line and θ is the angle between the shot's trajectory and the positive x-axis. Note that this value of θ is slightly different to that explained in the xG metric which measures the angle between the shot and the goal line (i.e. $90-\theta$).

10.3 Considering Distance Travelled to Games

The introduction of the Reduced Noise Feature Vector saw the removal of the location feature, which represented whether a team played at home/away (since the prediction was now implicitly for the home team only).

The idea of including the location is however still useful if considering the distance travelled to the game: for a home team, this would be set to 0; for an away team, this would be the direct distance between arenas. Through analysis of the predictions, a question that could potentially be answered is whether the distance travelled by a team influences their probability of winning. This could potentially negatively influence the away team due to the likelihood of being tired from travelling and the possibility of muscle cramp from restricted body movement on their method of transport.

This question arose during the analysis of the 2021/22 season games where 53.7% of games were won by the home side (Figure 16). If distance had no impact on game outcome, then this statistic is expected to be closer to 50%. The extra 3.7% corresponds to 48 more games being won by the home side; this is a considerable number throughout the course of one season, therefore warrants further investigation.

10.4 Betting Model

The success of the 2021/22 season prediction also leads to the possible implementation of a betting model. For each game of a season, the odds of betting solely on the predicted winner can be obtained from multiple companies, with the odds providing the greatest return from this set chosen. A simulated $1 bet can then be placed using this betting slip. The ground truth can then be obtained to determine whether this bet would win or lose.

Repeating this process for every game of the season can result in evaluating the Return On Investment (ROI) at the end. If this is positive, then the simulation has made money. This also suggests that the model's predictive ability is more powerful than those used by the betting companies, since lower odds are usually offered for the team they predict to win; this is to ensure the company does not lose too much money to winning bets.

This provides a further evaluation on the model's performance by also helping to quantify the usefulness of its applications to real world scenarios. The potential uses of the model helps to justify the importance of the project and the benefits that it provides to the hockey community.

10.5 Player Correlation

During the data collection phase, multiple statistics were collected on individual player performances such as their shifts, individual shots, goals etc. This data could be utilised to determine some form of player correlation metric which calculates the expected goals for a given line.

The analysis could be restricted to focus solely on one team in one season. The performance of a team is directly related to the performance of the skaters, therefore it is necessary for lines to work well together. Naturally however, some lines may play better than others. This can be investigated by analysing player data while the same set of players simultaneously skate a shift on the ice.

Trends may be identified by various regression models relating to the number of goals scored while a certain line is on the ice. These could utilise features similar to those used on a whole-team basis in this project. Other potential features could involve the use of data from the play immediately preceding a goal. This could highlight potential strategies that lead to the best scoring opportunities; lines that exploit these more are therefore more likely to score an increased number of goals.

11 Conclusion

Several conclusions have been derived from the exploration of multiple methods and questions in this project.

Through feature engineering, several feature vectors were developed in an attempt to maximise model accuracy. Using data from every NHL regular season game between the 2013/14 - 2020/21 seasons, we have successfully increased the accuracy of models from the benchmark of 59.38% by Weissbock et al. [1] to 62.7% with the use of the one vector per game implementation of the Basic Feature Vector and Stochastic Gradient Descent. This shows that further feature engineering can improve model accuracy but at a slow rate.

Restricting the training data to the 2021/22 season saw an increase in accuracy to 69.2% with the Reduced Noise feature vector and Neural Networks. This suggests that historical performances have little effect on the outcome of future games. NHL seasons can therefore be considered as independent, with multi-season datasets having decreased applicability compared to other sports.

The maximum accuracy achieved by our models however is still lower than those provided by models in other sports such as football [45]. This supports the proposed idea that hockey games are generally harder to predict, therefore will have a lower predictive accuracy regardless of the chosen model. The accuracy may also have an upper limit that cannot be surpassed with the use of mathematical models alone. This is most likely due to the 'luck' factor associated with the sport. Despite the attempt to quantify this feature with the PDO statistic and shot counts of deflections and tip-ins, it is concluded that luck cannot be easily measured and remains an elusive nuance of the sport that should be explored further.

When simulating the 2021/22 season with the trained models, we found that it was possible to predict the final standings of teams within each Division by ± 1 position for the majority of teams a whole season in advance. Points prediction however is less accurate with teams at the extremes of the table being significantly over/underpredicted wins.

It is possible to improve the accuracy of both standings and predicted points by reducing the simulation to the remaining games of a season currently in progress. Predicting the last 20% of the 2021/22 season resulted in 100% accuracy of the predicted standings of the Pacific Division. This is also a useful tool for coaching staff to utilise during a season to determine the amount of progress required by the team to secure Playoff qualification.

By extracting the 2019/20 and 2020/21 seasons from the dataset, we determined that the COVID-19 pandemic did have an external influence on game outcome that cannot be captured by the models (and therefore are less predictable). This was seen with an accuracy of 61.0% for models only considering the COVID-19 seasons compared to an accuracy of 62.7% for models excluding these seasons.

The 2019/20 and 2020/21 seasons should therefore be considered as outliers and excluded from future multi-season datasets.

Finally, future work has been proposed which could further extend the project and highlights some of its potential applications in the real world. Examples of this are extending the regular season simulation to focus on the Stanley Cup Playoffs or a betting model that uses Return on Investment (ROI) as another metric to evaluate the original model performance compared to those used by betting companies.

References

[1] J. Weissbock, H. L. Viktor, and D. Inkpen, "Use of performance metrics to forecast success in the national hockey league," in *MLSA@ PKDD/ECML*, pp. 39–48, 2013.

[2] S. Green, "Assessing the performance of premier league goalscorers." `https://www.statsperform.com/resource/assessing-the-performance-of-premier-league-goalscorers/`, 2012. Accessed 6/10/22.

[3] K. N. Jain, "Football analytics: A novel approach to estimate success," September 2022.

[4] W. Gu, K. Foster, J. Shang, and L. Wei, "A game-predicting expert system using big data and machine learning," *Expert Systems with Applications*, vol. 130, pp. 293–305, 2019.

[5] NHL Records, "Nhl history." `https://records.nhl.com/history`. Accessed 15/04/23.

[6] NHL, "Guide to 2013-14 nhl realignment." `https://www.nhl.com/news/guide-to-2013-14-nhl-realignment/c-685005`. Accessed 15/04/23.

[7] T. Campbell and A. Kimelman, "Nhl teams in new divisions with realignment for 2020-21 season." `https://www.nhl.com/news/nhl-teams-in-new-divisions-for-2020-21-season/c-319844882`. Accessed 25/11/22.

[8] Vegas Golden Knights, "Introducing the vegas golden knights." `https://www.nhl.com/goldenknights/news/introducing-the-vegas-golden-knights---newest-nhl-franchise/c-283997132`. Accessed 15/04/23.

[9] N. J. Cotsonika, "Kraken officially join nhl after final expansion payment." `https://www.nhl.com/news/seattle-officially-joins-nhl-can-sign-free-agents-make-trades/c-324191506`. Accessed 15/04/23.

[10] Arizona Coyotes, "Coyotes announce team name will change to arizona coyotes beginning in 2014-15." `https://www.nhl.com/coyotes/news/coyotes-announce-team-name-will-change-to-arizona-coyotes-beginning-in-2014-15/c-702881`. Accessed 15/04/23.

[11] P. Jensen, "Hedman wins hardest shot at 103.2 mph at all-star skills." https://www.nhl.com/news/hedman-wins-hardest-shot-at-all-star-skills/c-330523240, 2022. Accessed 26/11/22.

[12] K. Wesley, "Metrics for goal prediction in nhl ice hockey games: Progress report." November 2022.

[13] F. Azuaje, "Witten ih, frank e: Data mining: Practical machine learning tools and techniques 2nd edition," 2006.

[14] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Tech. Rep. MSR-TR-98-14, Microsoft, April 1998.

[15] I. Witten, "Decision trees." https://www.futurelearn.com/info/courses/data-mining-with-weka/0/steps/25391. Accessed 26/11/22.

[16] Python. https://www.python.org/. Accessed 09/10/22.

[17] "pandas." https://pandas.pydata.org/. Accessed 25/11/22.

[18] The Matplotlib development team, "Matplotlib: Visualization with python." https://matplotlib.org/. Accessed 18/11/22.

[19] scikit-learn. https://scikit-learn.org/stable/. Accessed 17/11/22.

[20] NHL, "Nhl stats api." https://statsapi.web.nhl.com/api/v1/. Accessed 23/11/22.

[21] ECMA, "Standard ecma-404 the json data interchange syntax," 2017.

[22] Microsoft, "Excel help & learning." https://support.microsoft.com/en-gb/excel. Accessed 10/10/22.

[23] K. Wesley, "Nhl game data 2013-2021." https://kaggle.com/datasets/4a62b280151cc5e72d0dad4b57778141792b25ac65829b2bb192610e7be61422. Accessed 17/04/22.

[24] Kaggle, "How to use kaggle." https://www.kaggle.com/docs/notebooks. Accessed 25/11/22.

[25] GitHub, "Github." https://github.com/. Accessed 10/10/22.

[26] R. Cunningham, "Batch Compute System, Department of Computer Science, University of Warwick." https://warwick.ac.uk/fac/sci/dcs/intranet/user_guide/batch_compute/. Accessed 24/11/22.

[27] K. Wesley, "Metrics for goal prediction in nhl ice hockey games: Specification." October 2022.

[28] monday.com. https://monday.com/. Accessed 12/10/22.

[29] M. Ellis, "Nhl game data." https://www.kaggle.com/datasets/martinellis/nhl-game-data?select=table_relationships.JPG, 2020. Accessed 15/11/22.

[30] G. Turi, "Online JSON Viewer." http://jsonviewer.stack.hu/. Accessed 23/11/22.

[31] M. Ellis, "Where the data comes from." https://www.kaggle.com/datasets/martinellis/nhl-game-data?select=table_relationships.JPG, 2021. Accessed 16/11/22.

[32] NHL, "Nhl to pause season due to coronavirus." https://www.nhl.com/news/nhl-coronavirus-to-provide-update-on-concerns/c-316131734. Accessed 16/04/23.

[33] XGBoost, "Xgboost documentation." https://xgboost.readthedocs.io/en/stable/. Accessed 14/02/23.

[34] scikit-learn, "Decision tree classifier documentation." https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. Accessed 10/01/23.

[35] scikit-learn, "Naïve bayes documentation." https://scikit-learn.org/stable/modules/naive_bayes.html. Accessed 10/01/23.

[36] scikit-learn, "Support vector classification documentation." https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Accessed 10/01/23.

[37] scikit-learn, "Multi-layer perceptron classifier documentation." https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. Accessed 10/01/23.

[38] scikit-learn, "Stochastic gradient descent documentation." https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html. Accessed 10/01/23.

[39] scikit-learn, "Adaboost classifier documentation." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html. Accessed 11/01/23.

[40] scikit-learn, "Gradient boosting classifier documentation." `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html`. Accessed 11/01/23.

[41] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, (New York, NY, USA), p. 208–211, Association for Computing Machinery, 2008.

[42] W. Jones, "How do hockey trades work in the nhl?." `https://hockeyanswered.com/how-do-hockey-trades-work-in-the-nhl/`. Accessed 18/04/23.

[43] NHL, "Positive test protocol." `https://media.nhl.com/site/asset/public/ext/2020-21/2020-21PositiveTestProtocol.pdf`, 2021.

[44] I. Kuitunen, M. M. Uimonen, and V. T. Ponkilainen, "Team-to-team transmission of covid-19 in ice hockey games – a case series of players in finnish ice hockey leagues," *Infectious Diseases*, vol. 53, no. 3, pp. 201–205, 2021.

[45] R. Bunker and T. Susnjak, "The application of machine learning techniques for predicting match results in team sport: A review," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1294–1298, 2022.