Building a Stanley Cup Winning Roster

Using a k-nearest neighbors algorithm

Lily Stensland

Project Summary

The goal of every NHL general manager is to construct a Stanley Cup winning roster. However, due to the limited availability of analytics tools, this task is extremely complex. In an attempt to simplify this process, a k-nearest neighbors algorithm was created, classifying players as one of four line-type players, using publicly available player evaluation statistics. The algorithm was modified to classify players as first liners, second liners, or bottom sixers, resulting in increased accuracy. Finally, the algorithm was modified to classify players as either top or bottom six players, leading to much higher accuracy. With further development, this algorithm opens the doors for increased confidence in player trades, drafting, and contract negotiations.

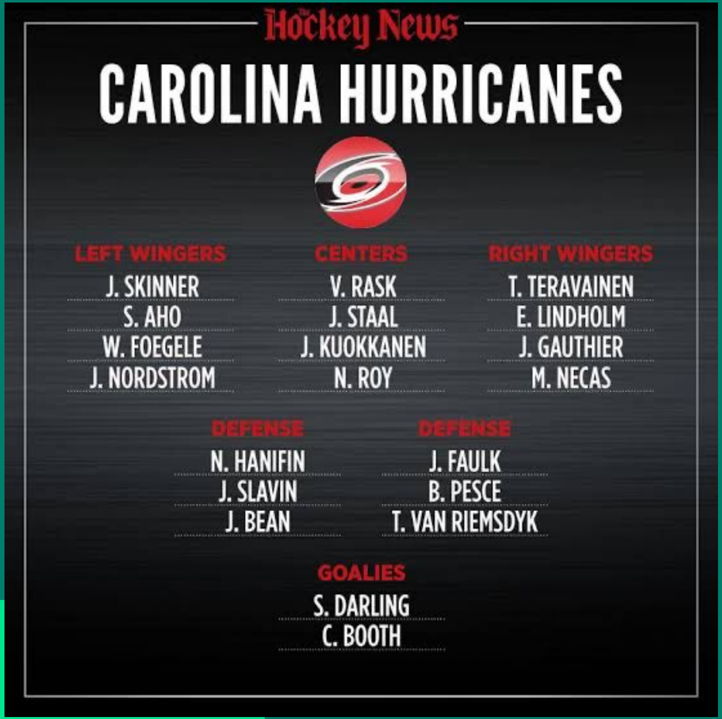
NHL Roster Building

NHL rosters are made up of 23 players:

- 12 forwards (4 lines of 3)
- 6 defensemen (3 pairs)
- 2 goaltenders
- 3 “healthy scratches”

A team’s best forwards make up the first line, with those players making the most money.

Teams must allocate limited cap space to pay all 23 players, while building the best team possible.



An example roster from the Carolina Hurricanes in 2020

The Problem

All 32 teams in the league are competing to create the best roster in order to win the Stanley Cup. However, a team cannot just sign the best 23 players. There exists a hard salary cap, which is meant to limit the spending by individual teams and increase league parity.

Now, teams are attempting to use hockey analytics to build the best possible teams under cap constraints. However, hockey remains very behind on data analytics investment and utilization.

The State of Hockey Analytics

FACT #1

Before 2014, there were only 15 data professionals employed by the NHL.

Today, most teams have large analytics departments and the league contracts multiple independent analytics firms.

FACT #2

Scouts and player evaluators relied on box scores and the eye test until rudimentary analytics began to be developed in the early 2000s.

FACT #3

Analysts have developed tools to measure shot quality, expected goals, and competition quality to create a fuller picture of a game beyond the box score.

A Classification Algorithm

Creating a Dataset

Pull skater-level season by season data from MoneyPuck.com

1. Collect the rosters of Stanley Cup winning teams from the past 10 seasons and classify each of the 12 forwards in lines according to ice time
2. Remove unwanted statistics and seasons of data from after a cup win
3. Aggregate the data to get one value for each desired statistic

Dataset Features

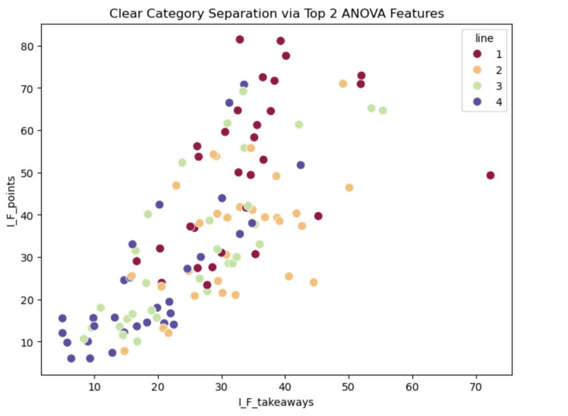
	name	line	onIce_xGoalsPercentage	onIce_corsiPercentage	onIce_fenwickPercentage	I_F_xGoals	I_F_shotsOnGoal	I_F_shotAttempts	I_F_points	I_F_g
0	Sidney Crosby	1	0.642222	0.611111	0.612222	24.567778	209.777778	349.222222	81.444444	31.444
1	Phil Kessel	1	0.563333	0.557778	0.548889	25.434444	266.000000	479.666667	64.666667	29.444
2	Nick Bonino	1	0.503750	0.483750	0.490000	8.871250	85.500000	141.875000	23.375000	9.375
3	Chris Kunitz	2	0.621111	0.607778	0.606667	18.362222	161.555556	281.555556	46.888889	20.555
4	Evgeni Malkin	2	0.645556	0.620000	0.617778	24.442222	214.666667	384.111111	71.000000	27.444
5	Patric Hornqvist	2	0.632222	0.595556	0.598889	21.663333	206.555556	313.777778	40.222222	19.333
6	Conor Sheary	3	0.610000	0.575000	0.570000	11.675000	103.500000	167.500000	31.500000	15.000
7	Scott Wilson	3	0.393333	0.440000	0.450000	5.340000	55.333333	91.333333	10.666667	4.333
8	Carl Hagelin	3	0.503333	0.508333	0.515000	14.013333	149.666667	248.500000	31.833333	13.000
9	Matt Cullen	4	0.516667	0.491111	0.492222	14.881111	129.333333	227.555556	35.444444	13.000
10	Bryan Rust	4	0.530000	0.513333	0.516667	7.700000	70.666667	116.666667	13.666667	6.666
11	Jake Guentzel	4	0.560000	0.540000	0.550000	10.740000	81.000000	136.000000	33.000000	16.000

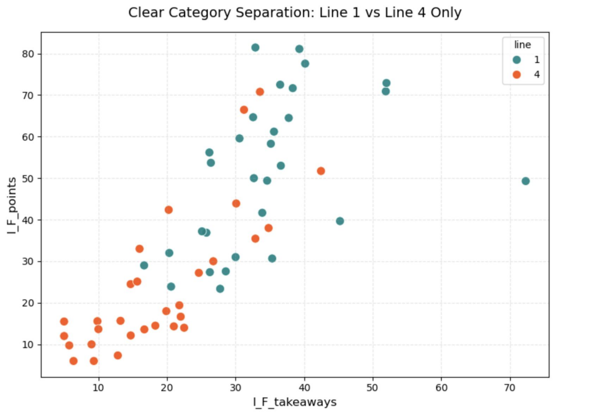
- **I_F_xGoals:** Expected Goals
- **I_F_shotsOnGoal:** Shots on goal
- **I_F_shotAttempt:** Shot attempts
- **I_F_points:** Goals + Assists
- **I_F_goals:** Goals
- **I_F_rebounds:** Rebound shot attempts
- **I_F_hits:** Number of hits a player has given
- **I_F_takeaways:** Number of takeaways the player has taken from opponents
- **I_F_giveaways:** Number of giveaways the player has given to other team
- **I_F_highDangerShots:** High Danger Shots (Higher than 20% xGoal Value)
- **onIce_xGoalsPercentage:** On Ice xGoals For / (On Ice xGoals For + On Ice xGoals Against)
- **onIce_corsiPercentage:** On Ice Shot Attempts For / (On Ice Shot Attempts For + On Ice Shot Attempts Against)
- **onIce_fenwickPercentage:** On Ice Unblocked Shot Attempts For / (On Ice Unblocked Shot Attempts For + On Ice Unblocked Shot Attempts Against)

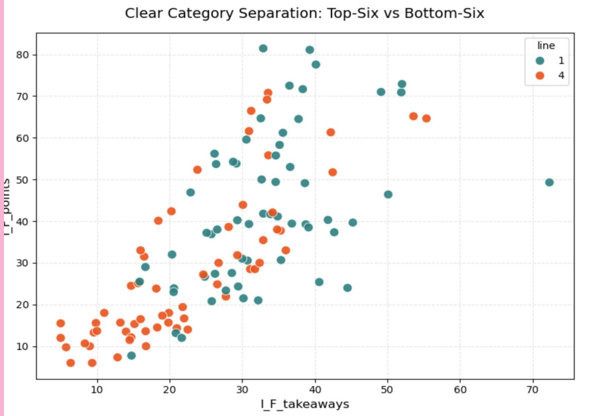
ANOVA F-Test

	Feature	F-Score	p-value
10	I_F_takeaways	12.848551	2.626479e-07
6	I_F_points	12.714543	3.043559e-07
3	I_F_xGoals	11.491071	1.187957e-06
7	I_F_goals	10.312484	4.536448e-06
12	I_F_highDangerShots	10.211760	5.093429e-06
4	I_F_shotsOnGoal	9.360785	1.366126e-05
5	I_F_shotAttempts	9.261434	1.534378e-05
11	I_F_giveaways	7.954008	7.208519e-05
8	I_F_rebounds	7.333552	1.520375e-04
1	onIce_corsiPercentage	5.305445	1.835214e-03
2	onIce_fenwickPercentage	5.252279	1.960956e-03
0	onIce_xGoalsPercentage	5.140822	2.253551e-03
9	I_F_hits	3.276315	2.359990e-02

With the top 2 ANOVA features, there is solid categorization for the first and fourth line, but the features struggle to differentiate between second and third line players







Create a k-nearest neighbors algorithm

Using the created dataset as training/testing data, create a k-nearest neighbors algorithm that classifies players into one of four lines.

Training on a dataset of Stanley Cup winning players should result in line predictions matching a player’s optimal usage, not necessarily their current.

Algorithm Creation

	precision	recall	f1-score	support
1	0.43	0.50	0.46	6
2	0.67	0.33	0.44	6
3	0.25	0.33	0.29	6
4	0.50	0.50	0.50	6
accuracy			0.42	24
macro avg	0.46	0.42	0.42	24
weighted avg	0.46	0.42	0.42	24

	precision	recall	f1-score	support
1	0.50	0.50	0.50	6
2	0.38	0.50	0.43	6
3	0.60	0.50	0.55	12
accuracy			0.50	24
macro avg	0.49	0.50	0.49	24
weighted avg	0.52	0.50	0.50	24

	precision	recall	f1-score	support
1	0.64	0.75	0.69	12
3	0.70	0.58	0.64	12
accuracy			0.67	24
macro avg	0.67	0.67	0.66	24
weighted avg	0.67	0.67	0.66	24

The k-nearest neighbors algorithm struggled to classify players into the third and fourth lines, likely due to the similar play styles of both lines.

By combining the third and fourth lines into a “bottom-six” class, the model had more success.

Combining the first and second lines into a “top-six” as well greatly increased the efficacy of the model.

Some Player Examples

Alex Ovechkin

Alex Ovechkin, by ice time, was a second line player in the 2025 to 2026 season.

All three versions of the algorithm predicted that Ovechkin should be a first line player.

While “Ovi” took a step back in his playing ability this past NHL season, largely due to age, he is well regarded as one of the best players of all time.

Because the algorithm considers Ovi’s stats from his entire, highly-awarded career, it predicts him to be a guaranteed first line player.

Some Player Examples

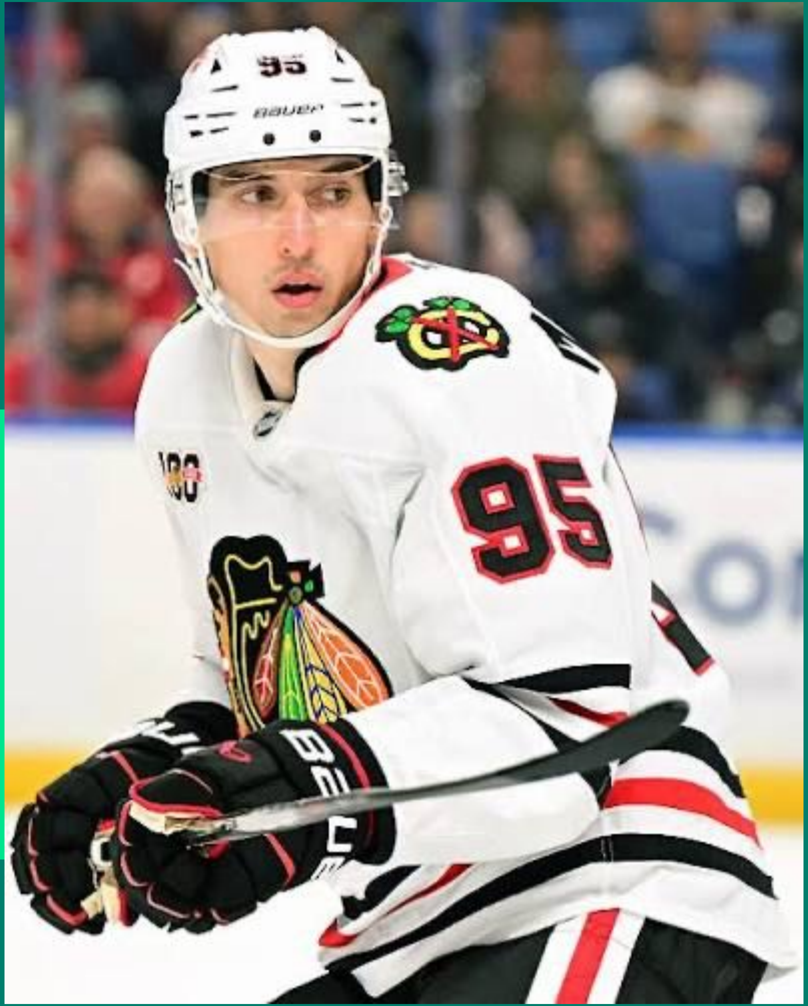
Ilya Mekheyev

Ilya Mekheyev, by ice time, was a first line player in the 2025 to 2026 season.

All three versions of the algorithm predicted that Mekheyev should be a third line/bottom six player.

Mekheyev plays for the Chicago Blackhawks, who were the second worst team in the NHL.

A bad team forces players to play higher in the lineup. However, if Mekheyev was traded to a Stanley Cup-caliber team, he should be moved much further down in the lineup.



Some Player Examples

Vladimir Tarasenko

Vladimir Tarasenko, by ice time, was a third line player in the 2025 to 2026 season.

The first algorithm predicted Tarasenko to be a second line player, while the other two algorithms predicted him to be a bottom-six player.

As a third-line player on a cup contending team (the Minnesota Wild), Tarasenko should be playing a line as predicted by the algorithm.

Tarasenko is a solid middle six player, which is reflected by the algorithm.

Evaluation of the Algorithm

Two evaluation methods

Near Winners

Run the rosters of the runner-up teams each year. If the algorithm predicts the lines of these rosters, then it is an effective model.

Changed Winners

Find players who changed lines immediately before winning a cup. If their pre-win statistics lead the algorithm to predict their winning lineup position, the model is effective.

Evaluation Datasets

Near Winners

This dataset takes the rosters of the Stanley Cup losers from the last 5 seasons:

1. 2021-2022 Tampa Bay Lightning
2. 2022-2023 Florida Panthers
3. 2023-2024 Edmonton Oilers
4. 2024-2025 Edmonton Oilers
5. 2025-2026 Vegas Golden Knights

name	team	line	onIce_xGoalsPercentage	onIce_corsiPercentage	onIce_fenwickPercentage	I_F_xGoals	I_F_s
Steven Stamkos	22 TBL	1	0.584286	0.572857	0.568571	24.566429	
Nick Paul	22 TBL	1	0.442857	0.498571	0.521429	6.284286	
Alex Killorn	22 TBL	1	0.546000	0.544000	0.541000	16.832000	
Anthony Cirelli	22 TBL	2	0.502000	0.490000	0.490000	13.068000	
Ondrej Palat	22 TBL	2	0.545000	0.534000	0.532000	14.535000	
Brandon Hagel	22 TBL	2	0.513333	0.486667	0.493333	8.856667	
Brayden Point	22 TBL	3	0.601667	0.573333	0.573333	24.175000	
Corey Perry	22 TBL	3	0.591429	0.567857	0.565714	24.457143	

Changed Winners

This dataset takes a selection of players who played on a different line the year before they won the cup than the year they won.

name	og_team	win_team	og_lines	win_lines	onIce_xGoalsPercentage	onIce_corsiPercentage	onIce_fenwickPercenta
Jordan Martinook	25 CAR	26 CAR	1	3	0.485455	0.505455	0.507
Taylor Hall	25 CAR	26 CAR	1	2	0.561333	0.567333	0.5620
Logan Stankoven	25 CAR	26 CAR	2	1	0.630000	0.600000	0.5950
Andrei Svechnikov	25 CAR	26 CAR	3	2	0.627143	0.632857	0.618
Jackson Blake	25 CAR	26 CAR	4	1	0.575000	0.630000	0.6300
Nikolaj Ehlers	25 WPG	26 CAR	3	2	0.569000	0.578000	0.5760
Aleksander Barkov	23 FLA	24 FLA	2	3	0.570000	0.570000	0.5650
Nick Cousins	23 FLA	24 FLA	3	4	0.526667	0.528889	0.531
Vladimir Tarasenko	22 NYR	24 FLA	4	3	0.570000	0.575455	0.578
Kyle Okposo	23 BUF	24 FLA	2	4	0.540000	0.537333	0.534
Jonathan Marchessault	22 VGK	23 VGK	1	2	0.588889	0.566667	0.554
Nicolas Roy	22 VGK	23 VGK	2	3	0.524000	0.534000	0.5220

Near Winners

This dataset reported low accuracy for the algorithms which predicted:

- 1st, 2nd, 3rd, and 4th lines (accuracy = 35%)
    - Struggled to classify the 3rd and 4th lines
- 1st and 2nd lines with a bottom-six (accuracy = 42%)
    - Tended to predict more players as first liners than in the set

However, there was pretty high accuracy for the algorithm which predicted a top and bottom six (accuracy = 65%)

- More often overestimated a player's talent

```
------- 1234 Prediction -------
Overall Accuracy: 0.3500

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.43      0.60      0.50        15
           2       0.47      0.47      0.47        15
           3       0.15      0.13      0.14        15
           4       0.27      0.20      0.23        15

    accuracy                           0.35        60
   macro avg       0.33      0.35      0.34        60
weighted avg       0.33      0.35      0.34        60

--- Confusion Matrix ---
[[9 2 2 2]
 [3 7 4 1]
 [4 4 2 5]
 [5 2 5 3]]
```

```
------- 123 Prediction -------
Overall Accuracy: 0.4167

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.28      0.47      0.35        15
           2       0.33      0.20      0.25        15
           3       0.58      0.50      0.54        30

    accuracy                           0.42        60
   macro avg       0.40      0.39      0.38        60
weighted avg       0.44      0.42      0.42        60

--- Confusion Matrix ---
[[ 7  1  7]
 [ 8  3  4]
 [10  5 15]]
```

```
------- 12 Prediction -------
Overall Accuracy: 0.6500

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.62      0.77      0.69        30
           3       0.70      0.53      0.60        30

    accuracy                           0.65        60
   macro avg       0.66      0.65      0.65        60
weighted avg       0.66      0.65      0.65        60

--- Confusion Matrix ---
[[23  7]
 [14 16]]
```

Changed Winners

This dataset reported much higher accuracy for all three algorithms.

Predicting 1st, 2nd, 3rd, and 4th lines (accuracy = 62%)

- Some struggles with identifying 3rd liners

Predicting 1st and 2nd lines with a bottom six (accuracy = 79%)

- Some struggle identifying 1st liners

Predicting top and bottom six (accuracy = 90%)

- Very few misattributions

```
------- 1234 Prediction -------
Overall Accuracy: 0.6207

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.50      0.60      0.55         5
           2       0.88      0.78      0.82         9
           3       0.50      0.44      0.47         9
           4       0.57      0.67      0.62         6

    accuracy                           0.62        29
   macro avg       0.61      0.62      0.61        29
weighted avg       0.63      0.62      0.62        29

--- Confusion Matrix ---
[[3 1 1 0]
 [1 7 1 0]
 [2 0 4 3]
 [0 0 2 4]]
```

```
------- 123 Prediction -------
Overall Accuracy: 0.7931

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.57      0.80      0.67         5
           2       0.88      0.78      0.82         9
           3       0.86      0.80      0.83        15

    accuracy                           0.79        29
   macro avg       0.77      0.79      0.77        29
weighted avg       0.81      0.79      0.80        29

--- Confusion Matrix ---
[[ 4  0  1]
 [ 1  7  1]
 [ 2  1 12]]
```

```
------- 12 Prediction -------
Overall Accuracy: 0.8966

--- Classification Report ---
              precision    recall  f1-score   support

           1       0.87      0.93      0.90        14
           3       0.93      0.87      0.90        15

    accuracy                           0.90        29
   macro avg       0.90      0.90      0.90        29
weighted avg       0.90      0.90      0.90        29

--- Confusion Matrix ---
[[13  1]
 [ 2 13]]
```

Conclusion

The algorithms were relatively successful in predicting player lines, especially when combing lines with similar identities to create a top and bottom six.

There remains work to be done in the future. This algorithm could be refined so that predicting all four lines has higher accuracy, perhaps by doing further analysis on the dataset features or by creating new offensive and defensive player evaluation metrics.

This algorithm could be modified to predict prospect NHL success, perhaps by tracking the pre-NHL statistics of proven NHL talents.

This algorithm could one day be used by NHL teams to scout trade and draft targets, and to aid in contract negotiations to ensure both cap compliance and a talented roster capable of winning a Stanley Cup.

Project Timeline

~~Week 1-2~~
~~Project Proposal~~

~~Week 3~~
~~Create the original dataset~~

~~Week 4~~
~~Create the k-nearest neighbors algorithm~~

~~Week 5~~
~~Create evaluation datasets~~

~~Week 6~~
~~Run datasets through algorithm and create evaluation metric~~

~~Week 7~~
~~Final report~~

Week 8
~~Final report,~~ peer reviews