Enhancing NHL Salary Evaluation through Dimensionality Reduction

Raphaël Fontaine
McGill University
Montreal, Canada
raphael.fontaine@mail.mcgill.ca

***Abstract*—This project explores the use of dimensionality reduction techniques to evaluate and analyze the salaries of NHL players. By applying methods such as PCA, PLS, Random Projection, and feature selection, the project aims to simplify the data while preserving important information. These techniques are assessed based on their ability to preserve model performance, reduce training time, and identify the most significant features in the dataset. While the models performed well, they struggled with edge cases, higher salaries, and non-statistical factors not captured in the dataset.**

Keywords—Dimensionality reduction, feature selection, hockey analytics

I. Introduction

Over the past decades, the rapid growth of data and the accessibility of datasets have influenced various aspects of our lives. Combined with refined artificial intelligence (AI) models, this transformation had an undeniable influence. The sports industry has taken advantage of this innovation by leveraging data to complement performance analysis, strategy, and decision-making [1].

Among all sports, ice hockey is one of the most intense, fast-paced, and complex. Over the years, the game has evolved on the ice, but also in its analysis. Data now plays a crucial role when developing strategies or evaluating player performance. The National Hockey League (NHL), the largest professional hockey league in the world, recently implemented a system that tracks the position of the players in real time. Along with other metrics, this data added a new dimension to the game. Teams are relying more and more on data to make informed decisions, and most organizations have established dedicated data analytics departments.

Each team in the NHL operates under a salary cap, meaning that the total of the salaries of all their players must not exceed a specific amount. Organizations consider numerous factors to determine the amount of a contract they offer, and data obviously plays an important role in this process. In addition to basic metrics, advanced statistics are also available to help teams in their decision-making. However, with that many available metrics, it is a considerable challenge to determine their importance.

This project explores methods to evaluate the salaries of NHL players and attempts to uncover the relevant information. The primary objective is to develop, explore, and compare various dimensionality reduction techniques. It also seeks to evaluate and assess the effectiveness of these techniques in simplifying the data while retaining critical information. Lastly, it aims to interpret the results and identify the most important features in the dataset.

Some related works in the field of sports analytics has demonstrated the growing importance of machine learning and dimension reduction to achieve similar tasks. For example, Mincev (2021) analyzed various methods to predict NHL player salaries [2]. Matsuzawa (2017) explored different machine learning models to predict future points in the NHL and applied dimensionality reduction to improve the results [3]. Additionally, Liu (2024) conducted a comparative study on model-agnostic interpretation frameworks for salaries in the National Basketball Association [4].

This project relates to the course material by highlighting the importance of feature engineering, a fundamental aspect of any AI model. The class material provided a brief introduction to Principal Component Analysis (PCA). This project builds on it and explore other dimensionality reduction techniques. Feature engineering can significantly enhance the accuracy and interpretability of models, which makes it an essential topic in the field AI.

II. METHODS

A. Dataset

Data from two sources was combined to create the dataset: *Spotrac* and *MoneyPuck*. *Spotrac* is one of the largest online resources for sports fantasy and contract information. *Moneypuck.com* is a hockey analytics website and offers several public datasets containing a range of basic and advanced statistics. The data required manual preprocessing to align the player and the team names between the two sources. Additionally, the salary cap in the NHL is increasing progressively. To account for this, salaries were adjusted and normalized to the 2023-2024 season, where the cap was $83.5M. This column serves as the label in the dataset and is recalled as simply the salary for the rest of the project. Categorical columns such as team, nationality, position, and handedness were encoded using one-hot encoding. Detailed instructions for replicating the dataset are available in the *README* file. The final dataset includes statistics from 11 seasons (2013 to 2023), for a total of 4020 records and 766 features. Each record represents the info and the statistics of a player for a specific season. The dataset is split by using the 2023 season for testing (439 records, ~11%) and the previous seasons for training (3581 records). Figure 1 below illustrates the density of the salaries and shows that most salaries fall between 1 and 6 million dollars.

The data is standardized by subtracting the mean and dividing by the standard deviation for each feature.

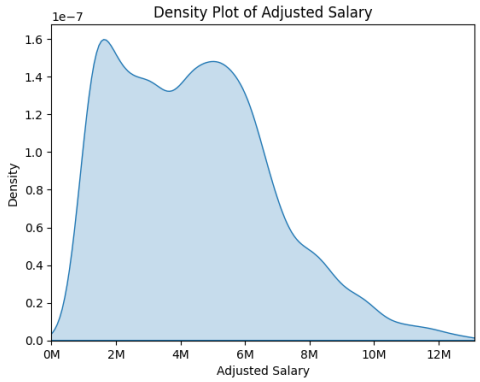

Fig. 1. Density Plot of Adjusted Salary

B. Models

Four models were selected based on their relevance in the related works mentioned earlier. These models were implemented using the *scikit-learn* library, as the focus of the project is to explore dimensionality reduction rather than to build AI models from scratch. Basic grid-search parameter tuning was first performed using the original dataset. The models were then trained and evaluated to create a baseline before applying dimensionality reduction. The selected models are Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR), and K-Nearest Neighbors (KNN). This approach enables the evaluation of dimensionality reduction on a variety of models.

C. Evaluation Metrics

To comprehensively evaluate each model's performance, six metrics are selected. Foremost, mean absolute error (MAE) serves as the primary metric, in order to capture the average absolute difference between salary evaluations and actual values. This metric is straightforward and interpretable, as it is expressed in the same unit (dollars) as the salaries. MAE is also calculated for the top 100 and top 50 highest salaries, emphasizing the performance on high-value salaries, which constitute a significant portion of the salary cap. The coefficient of determination (R^2) is also used to measure how well the model explains the variance in the target variable. Additionally, the symmetric mean absolute percentage error (SMAPE) is computed and evaluates the average percentage difference between predictions and actual values, providing a balanced measure that can handle wide target ranges. A smaller SMAPE value indicates better performance. Finally, the training time is included to assess the computational efficiency of the models.

D. Techniques

Dimensionality reduction refers to the process of reducing the number of features or variables in a dataset. It is important to understand the difference between feature selection and feature reduction. Feature selection simplifies the model by identifying and selecting the most relevant features and discarding redundant or irrelevant ones. On the other hand, feature reduction works by transforming the original features into a lower-dimensional representation. It attempts to combine features while keeping as much information as possible.

Like AI models, dimensionality reduction techniques can be unsupervised or supervised. Unsupervised techniques do not use the target variable in the reduction process. It aims to capture the underlying structure of the data itself. Conversely, supervised techniques try to leverage the relationship between the features and the target variable.

The selected techniques were chosen to explore various approaches and are detailed below. They are implemented from scratch and compared to the *scikit-learn* library implementations.

a) Principal Component Analysis (PCA):

PCA is an unsupervised feature reduction technique that transforms the original features into a smaller set of components. In fact, it combines potentially correlated features into what is called principal components [5]. PCA is implemented in two ways that should yield the same results: with Singular Value Decomposition (SVD) or by calculating the eigenvalues and eigenvectors of the covariance matrix. SVD in (1) is used to decompose X into three matrices: U, the left singular vectors, which correspond to the directions of maximum variance; S, the diagonal matrix of singular values; and V^T, the right singular vectors, which correspond to the principal components of the data.

$$X = USV^T \tag{1}$$

The next step is to calculate how much each component contributes to the total variance, called the explained variance. This is done using (2) where n is the number of samples. The value is then normalized using (3) to get the explained variance ratio over the total explained variance.

$$Explained\ Variance_i = \frac{s_i^2}{n-1} \tag{2}$$

$$Explained\ Variance\ Ratio_i = \frac{Explained\ Variance_i}{\sum s_i^2/(n-1)} \tag{3}$$

Finally, the components are ordered in ascending order of ratio to calculate the cumulative explained variance. Per example, the first component might explain 40% of the variance, the second component might add another 20%, and the cumulative sum would show that the first two components together explain 60% of the total variance. Different cumulative variance thresholds are set to evaluate the efficiency of PCA. V^T is also used to compute and interpret the important features of the components. The second PCA implementation works by first computing the covariance matrix C with (4). This matrix describes how the different features in the data relate to each other.

$$C = \frac{X^T X}{n-1} \tag{4}$$

The eigenvalues (λ) and eigenvectors (v) of C are determined with (5). The eigenvalues (λ) correspond to the explained variance and the ratio can be calculated with (6). The eigenvectors (v) are used to interpret the contribution of the features in each component.

$$Cv_i = \lambda_i v_i \tag{5}$$

$$Explained\ Variance\ Ratio_i = \frac{\lambda_i}{\sum \lambda_i} \tag{6}$$

b) Random Projection:

Random Projection (RP) is another unsupervised dimensionality reduction technique, and it projects high-dimensional data into a lower-dimensional subspace using a random matrix. RP is based on the Johnson-Lindenstrauss Lemma that states that a set of points in a high-dimensional space can be embedded into a space of much lower dimension while approximately preserving the pairwise distances between the points [6]. The "safe" minimal number of components, denoted as *k*, is defined in (7), where $0 < \epsilon < 1$. A small ϵ favors the accuracy of distance preservation while a larger ϵ allows more approximation [7]. Multiple values of ϵ are tested to evaluate this trade-off.

$$k \geq 4 \ln\left(n_{samples}\right) \left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \tag{7}$$

The features matrix X can then be projected (8) onto Gaussian random matrices G (9) or onto random sparse matrices R (10). The value of p is equal to the density $1/\sqrt{n_{features}}$. The result is a lower dimension matrix X of size (k, $n_{samples}$), that can be used to train and evaluate the models.

$$X_{lower} = XG \text{ or } XR \tag{8}$$

$$G_{ij} \sim N\left(0,\ \frac{1}{\sqrt{k}}\right) \tag{9}$$

$$R_{ij} = \begin{cases} -\sqrt{\frac{1}{k \cdot p}} & \frac{p}{2} \\ 0 \quad with\ probability & 1-p \\ +\sqrt{\frac{1}{k \cdot p}} & \frac{p}{2} \end{cases} \tag{10}$$

c) Partial Least Square (PLS):

PLS is a supervised feature reduction technique that aims to find the latent components that maximize the covariance between the features and the target variable. The algorithm works iteratively by repeating the following steps for every component to be generated [8]:

1. Find the directions (weights w) that explain the most covariance between X and y.
2. Project the data onto these directions to extract components t.
3. Use the components to compute loadings P for the features and loadings Q for the target.
4. Deflate X and y to remove the component's influence.

The full pseudo-code is available in the Appendix (Fig. 9). Finally, the original features can be projected onto the obtained weights to get the reduced features space. Again, this method is evaluated for different numbers of components.

d) Feature selection:

In addition to the previous methods, feature reduction techniques are explored. The first uses variance thresholds and assumes that features with low variance are less informative for the predictions. The second employs univariate selection, which consists of identifying the correlation between the features and the target variable. The Pearson correlation [9] is computed with (11). In both methods, the features with the lowest variance or correlation are discarded and tested with various thresholds.

$$R = \frac{(X - X_{mean})^T (y - y_{mean})}{(n-1)\, X_{std}\, y_{std}} \tag{11}$$

Ultimately, random feature sets are selected for comparison purposes, and curated selections of 10, 20, 30 and 40 features are used to assess the model performance relative to other dimensionality reduction methods.

III. Results

This section presents the results of the baseline, and the different dimensionality reduction approaches, along with some case studies. All the methods showed results similar to the implementation provided by the *sci-kit learn* library. In some cases, the implementations from scratch even performed slightly better than the libraries' implementation.

A. Baseline

For the baseline models using all the features, the MAE ranged between 1.3M and 1.4M. SVR achieved the lowest overall error, while LR performed the worst. This trend was consistent with the SMAPE values and R^2. In terms of training time, LR and KNN were very fast, while the other models required approximately 6 seconds for training. The detailed results are presented in Table 1 below.

TABLE I. Evaluation of the Baseline Models

Model	R^2	MAE	SMAPE	Train time (sec)
Linear Regression (LR)	0.5000	1,445,622	0.3712	0.62
Random Forest (RF)	0.5693	1,351,450	0.3462	6.82
Support Vector Regression (SVR)	0.5814	1,293,746	0.3263	5.56
K-Nearest Neighbors (KNN)	0.5568	1,352,233	0.3426	0.02

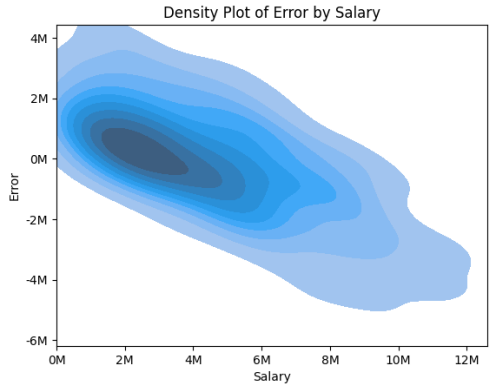

Fig. 2. Density Plot of Error by Salary, for SVR

The Top-100 and Top-50 MAE were worse than the overall MAE and highlight the models' difficulty to predict the highest salaries. Fig. 2 illustrates this phenomenon with a density plot of the error versus salary for the SVR model. A positive error indicates overestimation, while a negative error indicates underestimation. The model is notably accurate for salaries around 2-3M but is increasingly underestimating higher salaries.

B. Principal Component Analysis (PCA)

PCA produced promising results and demonstrated that the features can be effectively transformed into fewer components, without compromising too much accuracy. Both PCA approaches yielded the same results, validating the implementations. Various cumulative variance thresholds were investigated and showed that a balanced threshold is optimal. A very low threshold significantly reduces the number of components but loses too much information and negatively impacts the accuracy. A high threshold is effective to achieve great accuracy but reduces the benefits of dimensionality reduction. Fig. 3 depicts the number of components versus the cumulative explained variance threshold, showing the expected elbow-like pattern. For instance, the first 5 components explained around 65% of the total cumulative variance. The first principal component explained 37.14% of the variance alone and gave a lot of value to statistics of offensive actions in 5-on-5 situations, as seen in Appendix (Fig. 10).

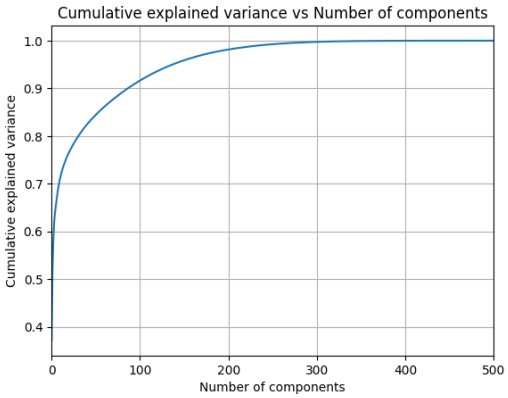

Fig. 3. Cumulative Explained Variance vs Number of Components

Even with as few as 17 components, the models maintained strong performance, as shown in Fig. 4 with a plot of the MAE by the number of components. Support Vector Regression even improved its baseline score when using 138 principal components, which is significantly less than the original 766 features. Surprisingly, Random Forest responded very badly to PCA compared to other models.

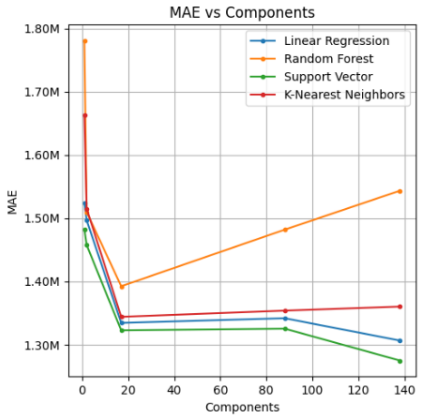

Fig. 4. MAE vs Number of Components for PCA

C. Random Projection

The safe number of components for projection was computed using multiple epsilon values, with results available in the Appendix (Table III). When epsilon was set to 0.9, as few as 202 components were needed, while 557 components were required with epsilon set to 0.4. Projecting the features onto Gaussian matrices was effective and had a minimal impact on accuracy. In fact, the MAE was under 1.4M for all models even with a projection to 202 components. It particularly benefited Linear Regression, who had the best predictions for the top salaries when using this method. The results for the projections to sparse matrices were slightly worse and led to a significant increase in training times for all models except K-Nearest Neighbors. Unfortunately, these methods offer no interpretability on the features and their importance.

D. Partial Least Square (PLS)

PLS achieved the best results, especially when heavily reducing dimensionality. With as few as 5 components, all models recorded MAE values of approximately 1.3M, with training times of around 1 second or less. With PLS, it is possible to measure the importance of the features in each component. The results differed from PCA, as the important features were not identical and were more balanced. Fig. 11 in the Appendix highlights the top features of the first component, where a lot of importance is given to goals and rebounds action. Notably, other less important components favored 4-on-5 statistics, ice time and position. Again, the performance of the Random Forest model deteriorated significantly when increasing the number of components, whereas the other models were consistent as shown in Fig. 5.

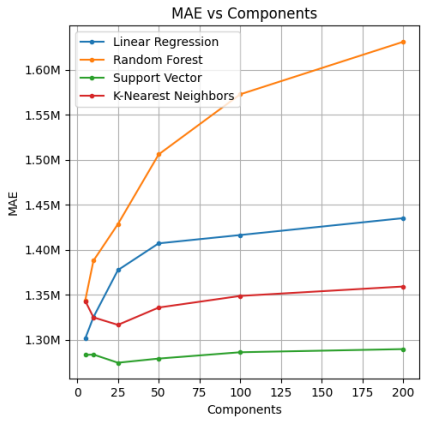

Fig. 5. MAE vs Number of Components for PLS

E. *Feature Selection*

The final set of techniques explored were feature selection methods. Removing features with the lowest variance did not improve the overall performance and led to worse results across all metrics when keeping a small number of features. However, there was a threshold at around 200 features where the models still performed well. The features with the highest variance were examined and a lot of them were not meaningful, such as whether the player is right-handed or left-handed.

Univariate selection showed more promise. For example, when selecting only the 32 most correlated features, all models achieved MAE values under 1.5M. Despite its higher training time, Random Forest stood out with its performance for predicting the top salaries with this technique.

Random reduced feature sets were also tested, yielding surprisingly good results. For example, with 100 features, Support Vector Regression achieved an MAE of 1,287,790 and Linear Regression scored 1,385,284 with only 20 features. In general, selecting random features was more successful that the two previous feature selection techniques.

Finally, curated lists of 10 to 40 features were tested. The results were satisfactory, with MAE values around 1.3M and 1.4M, even when using as few as 10 features. However, it did not beat previous techniques such as PCA and PLS.

F. *Case Study*

The final experiment consisted of using previous techniques to evaluate the salaries of selected players. Zach Hyman was first chosen as a case study. Three models were trained using all the data, excluding his: a baseline model with all features, and two reduced models using PCA and PLS with 3 components. Support Vector Regression (SVR) was selected due to its strong performance in earlier tests. Fig. 6 shows the actual salary, adjusted salary, and models predictions. The models were able to effectively estimate the salary and also track the increase in Hyman's salary after the 2020 season. By examining the important features of each component, it was found that PLS gives significance to a wide range of features, while PCA focuses on a smaller set of repeated metrics.

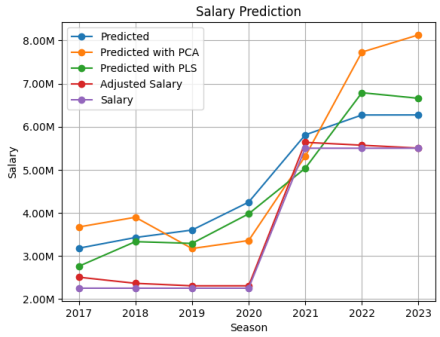

Fig. 6. Actual Salary and Predictions for Zach Hyman

The same process was applied to another player, Sidney Crosby. Both reduced models produced predictions with a similar shape to the baseline, as seen in Fig. 7. However, both methods consistently undershot Crosby's salary, with this phenomenon being more pronounced when using PCA. Crosby was also injured during the 2019 season and played only 41 games, but still posted impressive statistics. The predictions for that season were notably low, highlighting the inability of the models to account for this and generalize for these situations.

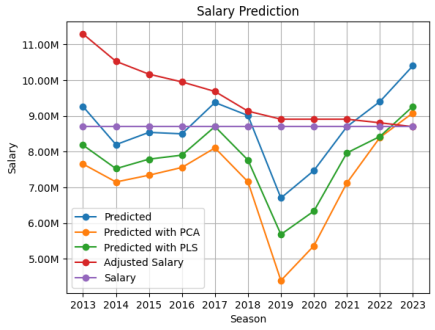

Fig. 7. Actual Salary and Predictions for Sidney Crosby

Interestingly, it is possible to take the first two components and create a scatter plot of all the salaries, with higher salaries represented in lighter colors. The result can be seen in Fig. 8, with only the salaries of the 2013 season displayed to increase readability. It is evident that the two components succeed to effectively categorize the data. However, when using the components 2 and 3 for the scatter plot, the categorization is less distinct, as shown in the Appendix (Fig. 12).

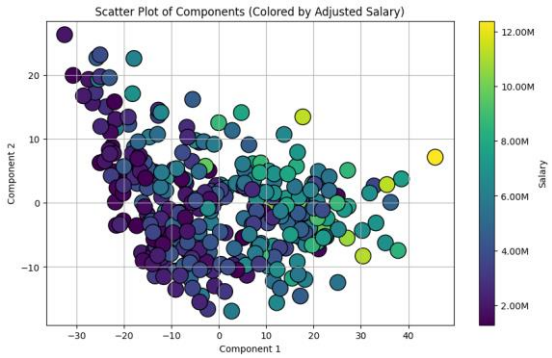

Fig. 8. Scatter Plot of Components 1 and 2 from PLS

IV. Discussion

This project demonstrates that dimensionality reduction techniques can effectively simplify a dataset while maintaining the performance of the model. Component Analysis (PCA) and Partial Least Squares (PLS) stood out as they were able to perform well with significantly fewer features. Both techniques offered interpretability as it was possible to inspect the main features contributing to the components. PLS highlighted meaningful hockey metrics, while PCA uncovered some interesting and unexpected results. Random Projection methods were successful in reducing dimensionality but faced limitations. The original features could only be transformed into a relatively high number of safe dimensions compared to other techniques. Additionally, this technique offers no insight into the importance of individual features and sometimes leads to longer training times.

Feature selection methods revealed the limitations of simply relying on variance or correlation. They achieved good results for a high number of features, but performance dropped drastically when reducing to a few features. Surprisingly, the random sets of features performed well, possibly indicating underlying patterns or redundancy in the dataset.

Overall, the models trained with the reduced feature sets showed comparable performance to those using the full dataset. Thus, it validates the hypothesis that dimensionality reduction can preserve essential information while simplifying the data.

A major success of this project was the significant reduction in training time achieved by applying dimensionality reduction techniques. The models could be trained much faster while maintaining similar performance. Furthermore, the implementations developed from scratch performed just as well as the pre-made methods. By using different thresholds, the project also revealed the trade-offs between the preservation of information and dimensionality reduction. Additionally, PCA and PLS successfully highlighted the contribution of the different features in the dataset and demonstrated their usefulness through some case studies. All those observations were particularly evident with the Support Vector Regression model, which achieved strong results across the project.

Undoubtedly, several key challenges emerged during the experiments. Feature selection methods were unable to outperform random feature sets, highlighting their limitations compared to other approaches. Moreover, the models failed to adapt to certain edge cases, such as injuries. The models succeeded to evaluate the salaries of many players. However, they consistently undershot the salaries of the higher-paid players. This trend can be explained by the fact that the models did not account for non-statistical elements. For instance, external factors like team needs, attachment to a team, or superstar status were not considered. These factors were not reflected in the dataset, and certainly play a crucial role in deciding the salaries in the NHL.

This project had noticeable limitations that affected its findings. The dataset was relatively small, and its generalization could benefit from a wider range of players and salaries. Inherently, the salary of a player in the NHL is not a "truth" value, as some players are obviously underpaid or overpaid. Salaries are also greatly influenced by factors outside the scope of this analysis, which limited the accuracy of the evaluations. The dataset lacked critical information about the players, such as contract duration, years with the same team, team salary distributions, and team or teammate statistics. These supplemental metrics could significantly improve the models and help with dimensionality reduction.

Future work should address these limitations by collecting a larger dataset with additional features. This would provide a more comprehensive and thorough analysis. There are many more dimensionality reduction techniques that were not explored. Additional methods or different AI models, such as deep learning, could provide further insights into the challenge and potentially improve accuracy and robustness. Finally, edge cases, such as injuries, could be accounted for with preprocessing or specialized models. More interpretable models are promising for future research and could certainly benefit hockey analysts and decision-makers.

References

[1] Morgulev, Elia, et al. "Sports analytics and the big-data era." *International Journal of Data Science and Analytics*, vol. 5, no. 4, Jan. 2018, pp. 213–22. https://doi.org/10.1007/s41060-017-0093-7.

[2] Mincev, Stepan. *Analysing Data Mining Methods in Sports Analytics: A Case Study in NHL Player Salary Prediction*, Universidade NOVA de Lisboa (Portugal), Portugal, 2021. *ProQuest*, https://proxy.library.mcgill.ca/login?url=https://www.proquest.com/dissertations-theses/analysing-data-mining-methods-sports-analytics/docview/2572307510/se-2.

[3] Matsuzawa, Takehiro. *Using Machine Learning to Predict Future Points in the NHL*. 7 Nov. 2017, nrs.harvard.edu/URN-3:HUL.INSTREPOS:37366468.

[4] Liu, Shun. "Model-Agnostic Interpretation Framework in Machine Learning: A Comparative Study in NBA Sports." *arXiv (Cornell University)*, Jan. 2024, https://doi.org/10.48550/arxiv.2401.02630.

[5] IBM. "What is principal component analysis (PCA)?" *IBM*, 8 Dec. 2023, www.ibm.com/topics/principal-component-analysis. Accessed 1 Dec. 2024.

[6] "Johnson–Lindenstrauss Lemma." *Wikipedia*, 11 Nov. 2024, en.wikipedia.org/wiki/Johnson%E2%80%93Lindenstrauss_lemma.

[7] Dasgupta, Sanjoy, and Anupam Gupta. "An Elementary Proof of a Theorem of Johnson and Lindenstrauss." *Random Structures and Algorithms*, vol. 22, no. 1, Nov. 2002, pp. 60–65. https://doi.org/10.1002/rsa.10073.

[8] "Partial Least Squares Regression." *Wikipedia*, 6 Nov. 2024, en.wikipedia.org/wiki/Partial_least_squares_regression.

[9] Berman, Jules J. "Understanding Your Data." *Elsevier eBooks*, 2016, pp. 135–87. https://doi.org/10.1016/b978-0-12-803781-2.00004-7.

APPENDIX

function PLS1(X, y, ℓ)
    $X^{(0)} \leftarrow X$
    $w^{(0)} \leftarrow X^{\mathrm{T}}y / \|X^{\mathrm{T}}y\|$, an initial estimate of w.
    for $k = 0$ **to** $\ell - 1$
        $t^{(k)} \leftarrow X^{(k)}w^{(k)}$
        $t_k \leftarrow {t^{(k)}}^{\mathrm{T}} t^{(k)}$ (note this is a scalar)
        $t^{(k)} \leftarrow t^{(k)} / t_k$
        $p^{(k)} \leftarrow {X^{(k)}}^{\mathrm{T}} t^{(k)}$
        $q_k \leftarrow y^{\mathrm{T}} t^{(k)}$ (note this is a scalar)
        if $q_k = 0$
            $\ell \leftarrow k$, **break the for loop**
        if $k < (\ell - 1)$
            $X^{(k+1)} \leftarrow X^{(k)} - t_k t^{(k)} {p^{(k)}}^{\mathrm{T}}$
            $w^{(k+1)} \leftarrow {X^{(k+1)}}^{\mathrm{T}} y$
    end for

Fig. 9. Pseudo-code for Partial Least Square [8]

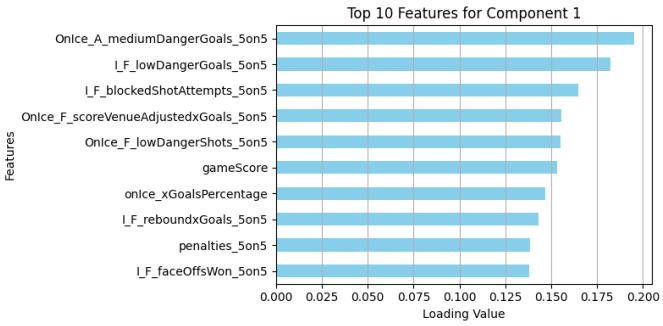

Fig. 10. Top 10 Contributing Features to Component 1 of PCA

TABLE II. EPSILON VALUE AND NUMBER OF COMPONENTS FROM THE JOHNSON-LINDENSTRAUSS LEMMA

Epsilon	Number of Components
0.1	7014
0.2	1888
0.3	909
0.4	557
0.5	392
0.6	303
0.7	250
0.8	219
0.9	202

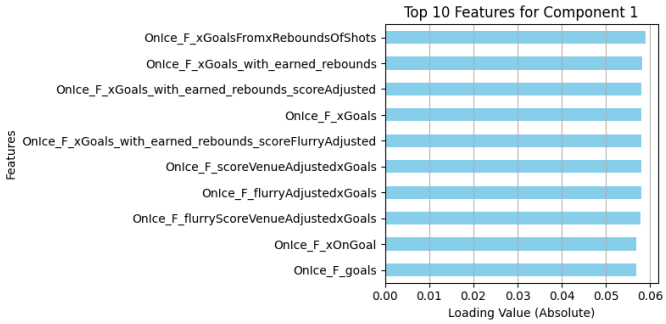

Fig. 11. Top 10 Contributing Features to Component 1 of PLS

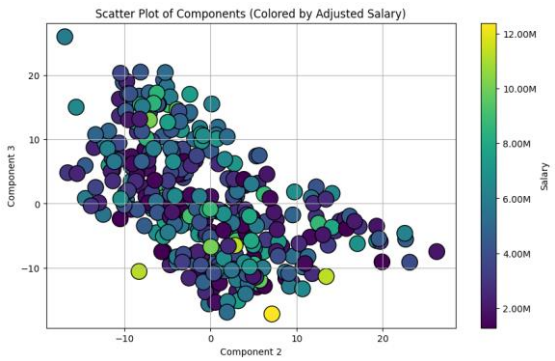

Fig. 12. Scatter Plot of Components 2 and 3 from PLS