



University of  
KwaZulu-Natal

November 2022

# COMP721

## Report: The National Basketball Association (NBA)

*Outlier Detection and Game Predictor*

Prepared by



Sashen Moodley (219006946)

# TABLE OF CONTENTS

|   |    |
|---|----|
| ABSTRACT.....                                     | 3  |
| 1 INTRODUCTION .....                              | 3  |
| 2 RELATED WORK .....                              | 3  |
| 3 METHODS AND TECHNIQUES.....                     | 4  |
| 3.1 OUTSTANDING PLAYERS – OUTLIER DETECTION ..... | 5  |
| 3.1.1 Data Pre-processing .....                   | 5  |
| 3.1.2 Data Analysis .....                         | 6  |
| 3.1.3 Dimensionality Reduction.....               | 6  |
| 3.1.4 Outlier Detection.....                      | 7  |
| 3.2 PREDICTING GAME OUTCOMES .....                | 10 |
| 3.2.1 Data Pre-processing .....                   | 10 |
| 3.2.2 Data Analysis .....                         | 11 |
| 3.2.3 Dimensionality Reduction.....               | 11 |
| 3.2.4 Game Predictions .....                      | 12 |
| 4 RESULTS AND DISCUSSION .....                    | 16 |
| 4.1 OUTSTANDING PLAYERS (OUTLIER DETECTION) ..... | 16 |
| 4.2 GAME PREDICTIONS .....                        | 17 |
| 5 CONCLUSION .....                                | 18 |
| REFERENCES .....                                  | 19 |

# ABSTRACT

---

Machine Learning has been employed across various domains. In recent years, there has been a growth in computational predictions for various sports. One of these sports that is of particular interest is The National Basketball Association (NBA). Due to its immense global popularity, there have been numerous initiatives to gather data about various players and game history, tracing back decades. This opulent source of data makes machine learning an appealing choice to detect outstanding players across and make predictions about the outcome of various games, amongst others. This report compares and contrasts various machine learning techniques in these two tasks. We utilise and evaluate various dimensionality reduction techniques (PCA and LLE), outlier detectors (SVM, IQR, Gaussian Mixtures), regressors (linear regression and polynomial regression), and a majority-voting ensemble of regressors. Through this evaluation we are able to conclude that the pairing of standard scaling LLE with the Gaussian Mixture anomaly was the best combination of techniques for detecting outstanding players. We also show that the ensemble technique is best for predicting game outcomes, with the strongest model in the ensemble being the polynomial regression model.

*The full system source code, along with a setup guide, is available via this [GitHub Link](#).*

## 1 INTRODUCTION

---

Machine Learning is the science (and art) of programming computers so they can learn from data [1]. Machine learning has proved to be a vital asset in today's society, as evident by its integration into various everyday systems, from email spam detection to self-driving cars. One area that has seen a recent boost in popularity is the use of machine learning for various sporting tasks. The National Basketball Association (NBA) is one of largest sports in the world, and is only increasing in popularity [2].

Along with the growth of these sports, there has been a movement into the era of Big Data [3]. Data has become ubiquitous and unavoidable in the modern era, with sporting being a major data-generator, having detailed records of various matches dating decades ago. Machine learning refers to a set of methods that can automatically detect patterns in data, and then use those uncovered patterns to predict future data or to perform other kinds of decision making under uncertainty. It therefore only natural for a marriage of machine learning and sports to perform a variety of tasks driven by data.

In this research, an empirical evaluation of various machine learning methods is conducted in two tasks: (1) outstanding player detection, and (2) game outcome predictions. An evaluation is made between two dimensionality reduction techniques (*Principal Component Analysis* – PCA – and *Locally Linear Embedding* – LLE), three outlier detectors (*Support Vector Machine* – SVM –, *Inter-Quartile Range* – IQR – and *Gaussian Mixtures*), two regressors (*Linear Regression* and *Polynomial Regression*), and an *ensemble* technique using the various regressors following a hard/majority voting strategy.

The rest of this report is structured as follows: Section 2 reviews related work in the field. Section 3 presents the employed methodology and the relevant comparisons between the machine learning techniques. These results are followed by a discussion in their appropriate subsection. Finally, Section 4 presents an overall summary of the results and discussion in prior section.

## 2 RELATED WORK

---

Machine learning has seen fruitful use various sporting domains such as volleyball [4], football [5], and ice hockey [6] to name a few. A sport that is of particular interest in this research is basketball, more specifically an analysis of machine learning techniques in the NBA-themed datasets.

Prior machine learning research in the NBA has a diverse scope. A surprisingly large portion of these tasks revolve around the social dynamics of the sport. In particular, [7] looks at implicit biases with regard to refereeing and uses machine learning to help predict one of three identified bias classes. Furthermore, [8] analyses offensive play calls using machine learning classification on prior instances where a referee viewed a foul play and under what conditions these were noted.

There has also been various evaluations of data mining and feature engineering for NBA datasets. [9] demonstrates how reporting on a single 'Elo' score for players finds better results in box-score predictors than multiple features. In

substantiating the historical richness of sporting data, [10] presents various feature extraction methods from decade-old datasets with the aim of their data mining to produce future knowledge discovery.

More closely related to this research, [11] conducted research into outlier detection within an NBA dataset. However, this was mostly done to demonstrate the effectiveness of ‘commute distance’ for similarity measures over Euclidean distance, rather than focussing on the results of their outlier detection.

There has also been a plethora of work in using NBA datasets to predict the outcomes of future games. [12] performs three predictive tasks, showing how distinctive statistical features outperform elementary ones in predicting playoff results. [12] additionally uses multiple linear regression in NBA “hot streak” with additive correlations between their experiments being made. [13] makes use of several machine learning methods including Naïve Bayes, artificial neural networks, and decision trees to make predictions on NBA games.

Whilst prior research has shown competency in demonstrating the efficacy of ‘final’ machine learning techniques, there has been limited research doing a joint comparison of various machine learning techniques for a single task. This research wishes to take a blend of the aforementioned approaches, by evaluating the efficacy of various machine learning techniques from data prepressing to making final predictions, and viewing the resultant effect that these various combinations have on one another – not isolating the results of each technique.

### 3 METHODS AND TECHNIQUES

The methodology employed throughout the development of this project, as well as the various comparisons made in each machine task is presented in Figure 1.

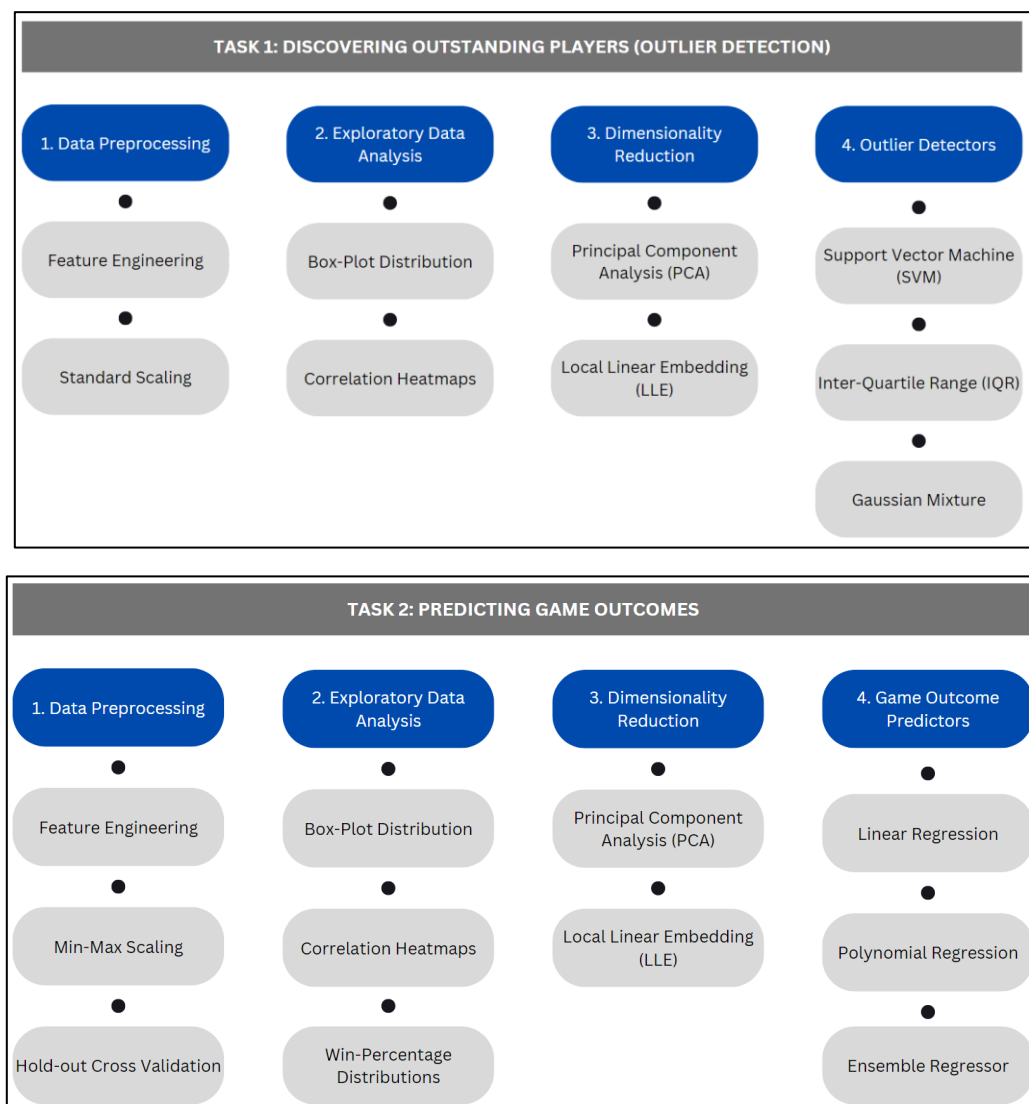


Figure 1: Visualization of the various machine learning techniques applied at each stage for each of the tasks

When comparing results, particularly for task 2 (game winner prediction) where I perform linear regression, I make use of comparative metrics. Namely, I present the *mean squared error* and the *R-2 score* for these regressors. During the report we make mention to various metrics. A formalism for these metrics is presented in Table 1.

Table 1: Metrics

| Metric           | Description  | Formula  |
|------------------|--|--|
| <b>RMSE</b>      | Root Mean Square Error - measure of the differences between values predicted by a model or an estimator and the values observed. | $RSME = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}}$ |
| <b>MSE</b>       | Mean Square Error - the average squared difference between the estimated values and the actual value.                            | $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$       |
| <b>R2 Score</b>  | Coefficient of determination.  | $R^2 = 1 - \frac{RSS}{TSS}$                                |
| <b>Precision</b> | Ratio of correctly predicted positive observations to the total predicted positive observations                                  | $P = \frac{TP}{TP + FP}$                                   |
| <b>Recall</b>    | Ratio of correctly predicted positive observations to all observations in actual class   | $R = \frac{TP}{TP + FN}$                                   |
| <b>F1-Score</b>  | Weighted average of Precision and Recall   | $F1 = 2 \times \frac{P \times R}{P + R}$                   |

### 3.1 OUTSTANDING PLAYERS – OUTLIER DETECTION

In this task, we are required to find out who are the outstanding players from the provided datasets. In particular, we perform outlier detection and pay attention to those “successful” outliers as opposed to low performing outliers. I will be using all the player datasets to conduct this outlier detection; however, I have grouped the players into two separate datasets: (1) yearly player stats and (2) player career stats. These respective categories contain both the regular and playoff season data, and are differentiated by their date of collection – yearly versus across their career.

The following machine learning methods were evaluated at each stage for this task:

- **Data pre-processing**
  - Feature engineering
  - Standard scaling
- **Exploratory data analysis**
  - Box-plot distributions
  - Correlation heatmaps
- **Dimensionality reduction**
  - Principal Component Analysis (PCA)
  - Local Linear Embedding (LLE)
- **Outlier detection**
  - Support Vector Machine (SVM)
  - Inter-Quartile Range (IQR)
  - Gaussian Mixture

#### 3.1.1 Data Pre-processing

As referenced in Section 2, data pre-processing plays an important role. For feature engineering the playoff and regular season datasets had to be concatenated. This involved swapping various columns and ensuring their formats were congruent. Furth more, I dropped out null-containing rows and columns that were not necessary for outliers; these were the non-numerical values hence there was no need for an extensive check of the discriminative qualities of each feature.

After this, I performed standard scaling on each feature. Standard scaling involves cantering our features around the mean with a unit standard deviation. This is an important step for our SVM and dimensionality reduction steps. The final version of these pre-processed datasets along with a comparison of their prior state is presented in Figure 2.

|   | id        | year | firstname | lastname  | team | leag | gp | minutes | pts | oreb | reb | ... | ast | blk | turnover | pf  | fga | fta | ftr | tpa | tpm |   |
|---|-----------|------|-----------|-----------|------|------|----|---------|-----|------|-----|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|---|
| 0 | ABRAMJ001 | 1946 | John      | Abramovic | PIT  | N    | 47 | 0       | 527 | 0    | 0   | 0   | 0   | 0   | 0        | 161 | 834 | 202 | 178 | 123 | 0   | 0 |
| 1 | ALBUCKH01 | 1946 | Chet      | Aubuchon  | CHI  | N    | 30 | 0       | 65  | 0    | 0   | 0   | 0   | 0   | 0        | 46  | 91  | 23  | 35  | 19  | 0   | 0 |
| 2 | BAKERNO01 | 1946 | Norm      | Baker     | CHI  | N    | 4  | 0       | 0   | 0    | 0   | 0   | 0   | 0   | 0        | 0   | 1   | 0   | 0   | 0   | 0   | 0 |
| 3 | BALTHED01 | 1946 | Herschel  | Baltimore | STL  | N    | 58 | 0       | 138 | 0    | 0   | 0   | 0   | 0   | 0        | 98  | 263 | 53  | 69  | 32  | 0   | 0 |
| 4 | BARRJO01  | 1946 | John      | Barr      | STL  | N    | 58 | 0       | 295 | 0    | 0   | 0   | 0   | 0   | 0        | 164 | 438 | 124 | 79  | 47  | 0   | 0 |

(a) – Yearly before

|           | id        | gp        | minutes   | pts      | oreb     | dreb      | reb       | ast       | stl       | blk       | turnover  | pf        | fga       | fgm       | fta       | ftr       | tpa       | tpm |
|-----------|-----------|-----------|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| ABRAMJ001 | 0.363215  | 0.959039  | 0.240012  | -0.14961 | 0.631968 | -0.780403 | 0.444668  | 0.659747  | -0.460339 | 0.616305  | 0.728461  | 1.289376  | 0.281827  | 0.450542  | 0.348543  | 0.374128  | -0.344244 |     |
| ALBUCKH01 | -0.151735 | -0.959039 | -0.273685 | -0.14961 | 0.631968 | -0.780403 | -0.562307 | -0.659747 | -0.460339 | -0.676205 | -0.556413 | -0.632958 | -0.718472 | -0.557716 | -0.460521 | -0.374128 | -0.344244 |     |
| BAKERNO01 | -1.173826 | -0.959039 | -0.849130 | -0.14961 | 0.631968 | -0.780403 | -0.721633 | 0.659747  | -0.460339 | -0.676205 | 1.070363  | 0.882208  | 0.648534  | 0.394493  | 0.760794  | -0.374128 | -0.344244 |     |
| BALTHED01 | 0.608363  | -0.959039 | -0.545171 | -0.14961 | 0.631968 | -0.780403 | -0.595021 | 0.659747  | -0.460339 | -0.676205 | 0.024873  | -0.217746 | 0.562523  | -0.317990 | -0.486613 | -0.374128 | -0.344244 |     |
| BARRJO01  | 0.608363  | -0.959039 | -0.234420 | -0.14961 | 0.631968 | -0.780403 | -0.254316 | -0.659747 | -0.460339 | -0.676205 | 0.226674  | -0.179342 | -0.267483 | -0.489350 | -0.374128 | -0.344244 |           |     |

(b) – Yearly After

|   | id        | firstname | lastname     | reg | gp   | minutes | pts   | oreb | dreb | reb   | ... | ast  | blk  | turnover | pf   | fga   | fta   | ftr  | tpa  | tpm  |
|---|-----------|-----------|--------------|-----|------|---------|-------|------|------|-------|-----|------|------|----------|------|-------|-------|------|------|------|
| 0 | ABDELAU01 | Alaa      | Abdelnaby    | N   | 256  | 5000    | 1465  | 283  | 563  | 846   | ... | 71   | 69   | 247      | 484  | 1236  | 620   | 321  | 225  | 3    |
| 1 | ABDULKA01 | Kareem    | Abdul-jabbar | N   | 1360 | 51466   | 38387 | 2975 | 5094 | 17440 | ... | 1160 | 3189 | 2527     | 4657 | 28307 | 15837 | 9504 | 6172 | 18   |
| 2 | ABDULMA01 | Mahmo     | Abdul-ma     | N   | 386  | 13633   | 8535  | 219  | 868  | 1087  | ... | 487  | 46   | 363      | 1107 | 7945  | 3514  | 1161 | 1051 | 1339 |
| 3 | ABDULSH01 | Tariq     | Abdul-shahid | N   | 236  | 4808    | 1830  | 275  | 501  | 776   | ... | 184  | 82   | 309      | 485  | 1726  | 720   | 529  | 372  | 76   |
| 4 | ABDURSH01 | Shaneel   | Abdur-shahid | N   | 672  | 24052   | 12338 | 1408 | 3976 | 5474  | ... | 718  | 356  | 1911     | 1345 | 10215 | 4729  | 4427 | 3614 | 477  |

(c) – Career before

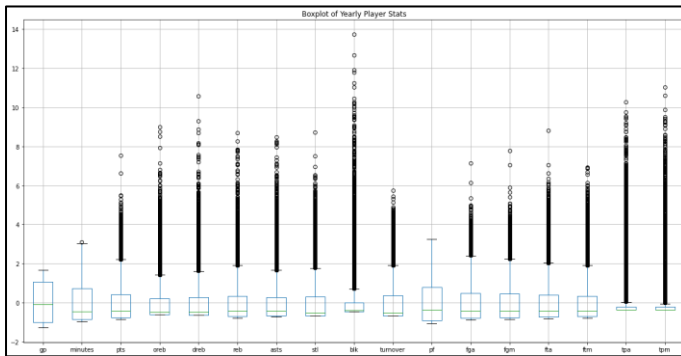
|           | id       | gp        | minutes   | pts      | oreb     | dreb      | reb       | ast       | ast       | blk      | turnover | pf        | fga       | fgm       | fta       | ftr       | tpa       | tpm |
|-----------|----------|-----------|-----------|----------|----------|-----------|-----------|-----------|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| ABDELAU01 | 0.333815 | -0.109599 | -0.089419 | 0.301975 | 0.202007 | 0.040599  | -0.343175 | -0.130106 | 0.026717  | 0.131053 | 0.097796 | -0.089307 | -0.049993 | -0.172907 | -0.187676 | -0.258906 | -0.249170 |     |
| ABDULKA01 | 5.450855 | 7.242396  | 10.140856 | 6.729470 | 9.620407 | 9.965913  | 5.762668  | 4.188157  | 14.735923 | 5.302706 | 6.135618 | 9.052088  | 10.896267 | 8.461610  | 7.999601  | 0.220904  | -0.242068 |     |
| ABDULMA01 | 1.630266 | 1.575568  | 1.875562  | 0.149167 | 0.527454 | 0.184748  | 1.840762  | 1.519478  | -0.081761 | 1.755134 | 0.993021 | 2.168956  | 2.031882  | 0.634506  | 0.854695  | 3.174178  | 3.117100  |     |
| ABDULSH01 | 0.257364 | 0.108349  | 0.011774  | 0.282874 | 0.133850 | -0.001270 | -0.144934 | 0.319798  | 0.088005  | 0.271686 | 0.099343 | 0.067335  | 0.020031  | 0.027024  | -0.000169 | -0.071036 | -0.121337 |     |
| ABDURSH01 | 1.967607 | 2.826465  | 3.202258  | 3.202544 | 3.643818 | 2.808729  | 1.586663  | 2.432474  | 3.226273  | 3.905453 | 2.066997 | 2.536992  | 2.548018  | 3.773806  | 4.089076  | 0.959313  | 0.767698  |     |

(d) – Career After

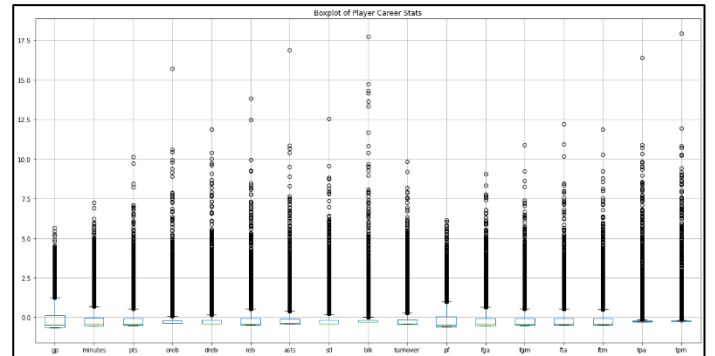
Figure 2: Data pre-processing output comparison

### 3.1.2 Data Analysis

The next step in my employed methodology is to perform some exploratory data analysis. This is in hopes of determining if the spread of the data supports the notion of any outliers prior to detecting them. To do this, I used two data analysis techniques to help manually detect these outliers, namely statistical box-plots and correlation heatmaps for each of the player datasets. The results of these techniques are presented in Figures 3 and 4.

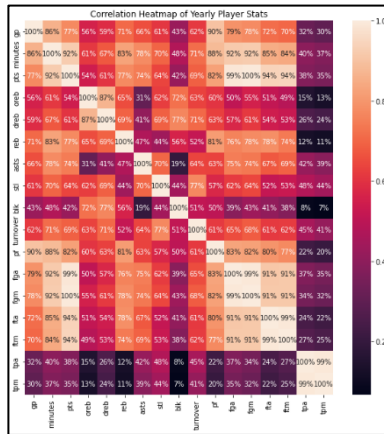


(a) – Yearly boxplots

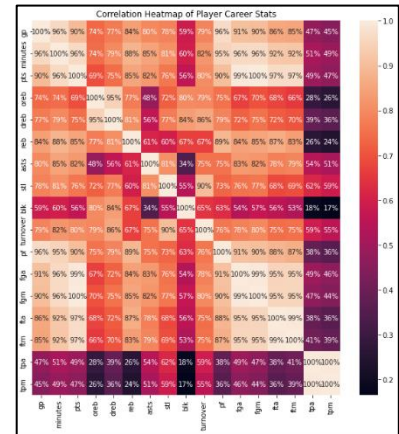


(b) – Career boxplots

Figure 3: Box plots for player statistics



(a) – Yearly correlation heatmap



(b) – Career correlation heatmap

Figure 4: Correlation heatmaps for player features

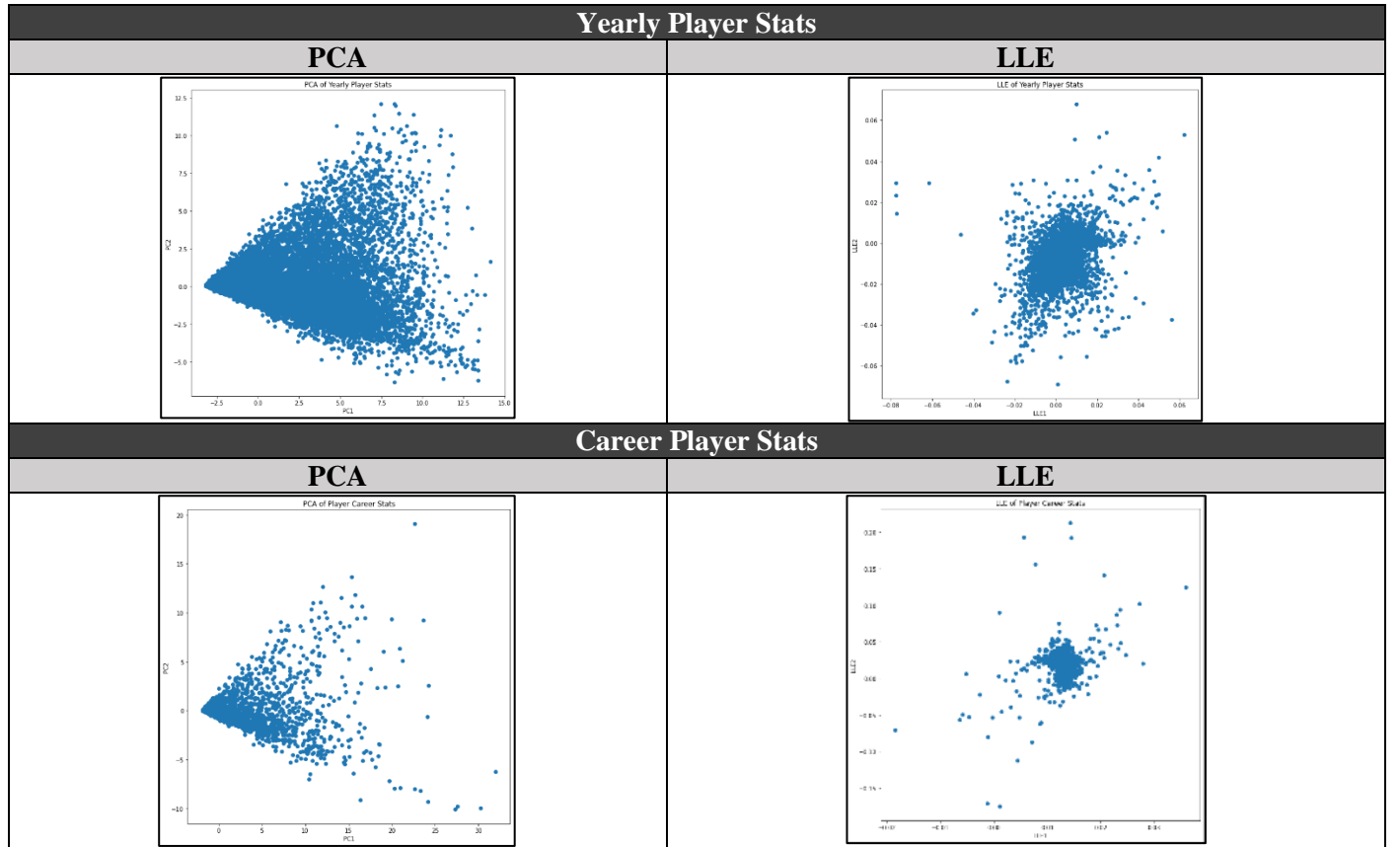
As we can see, the results of this step support the notion that there are outliers, particularly if we focus our attention on the box-plots. We can see that for certain player features, we have anomalies. Linking this finding back to the previous phase, of data pre-processing, we can see that the standard scaling did not ‘squash’ the players together, i.e., the scaler still preserves outliers whilst being essential for our future machine learning techniques.

### 3.1.3 Dimensionality Reduction

Even though we removed the non-numerical data from our player datasets, in the data pre-processing phase, we still have plenty of attributes related to player statistics. However, as we have shown in the data analysis stage, some of these features are not very discriminate. In an outlier detection context this means that they do not have enough variance in

their respective feature to account for the overall joint variance of the feature space. We can therefore perform dimensionality reduction to reduce our feature space, this improving time complexities, aiding visualization, and potentially increasing the chance to detect outliers whilst still having the original feature space representative of the original. In this phase we evaluate (compare and contrast) two machine learning dimensionality reduction techniques: Principal Component Analysis (PCA) and Local Linear Embedding (LLE). Both of these were set to reduce the dimensions down to ‘2’. The resulting lower dimensional feature spaces are presented in Table 2.

Table 2: Dimensionality reduction outcomes



As we can see both dimensionality reduction techniques provided us with different spreads for our data. This is because they leverage different paradigms for reduction. PCA uses a technique of projection whereas LLE uses manifold learning. A more detailed comparison of these techniques is provided at the end of this section, when we survey the results when taking all the techniques at each phase into consideration – making a joint comparison rather than treating the techniques in isolation. Therefore, rather than eliminating the ‘worst’ performing technique, we carry forward both into the next section.

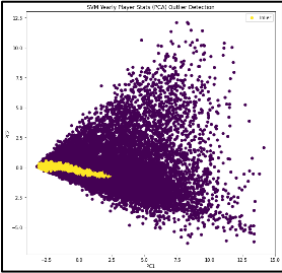
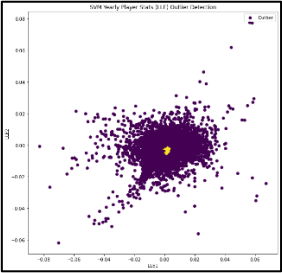
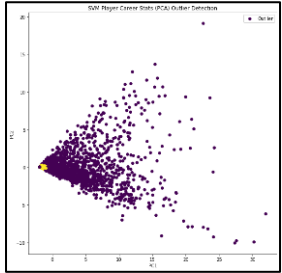
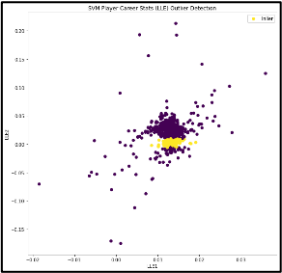
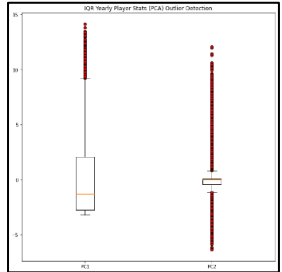
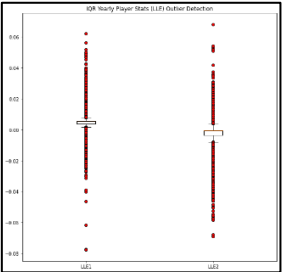
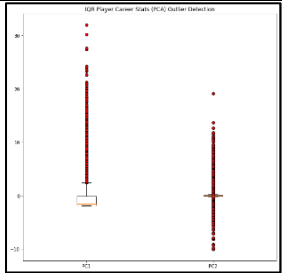
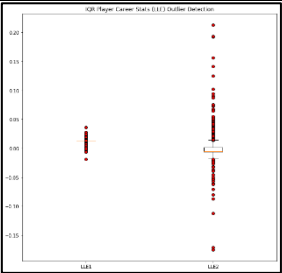
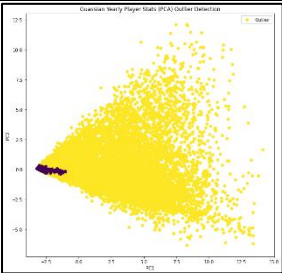
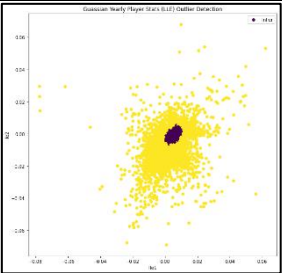
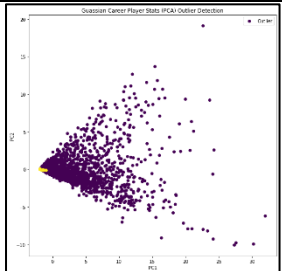
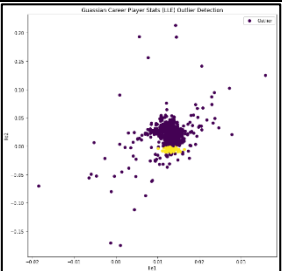
### 3.1.4 Outlier Detection

We now reach the final stage of our methodology for this task. The subsequent phases have allowed us to determine that there are indeed outliers in the dataset and helped us to expose them as much as possible for the machine learning techniques in this phase. Here we evaluate three techniques for detecting outliers: (1) Support Vector Machine, (2) Inter-Quartile Range, and (3) Gaussian Mixture.

Whilst the support vector machine might be most commonly associated with classification tasks, we can actually use it for outlier detection leveraging it’s *OneClass* variant provided in *scikit-learn*. Inter-Quartile range is a purely statistical approach to outlier detection and is probably the most intuitive technique as it works out the difference between the Q3 and Q1 spreads of data and projects a range above and below these respective thresholds to determine any outliers. Finally Gaussian Mixtures is a probabilistic model that assumes that the instances were generated from a mixture of several Gaussian distributions whose parameters are unknown [1]. Using this, we can detect and outlier (anomaly) as any point that is located in a low-density region.

Using these techniques, we perform outlier detection on the pre-processed, and reduced datasets and visualise their results across both dimensionality reduction techniques. This is presented in Table 3.

Table 3: Visualization of outliers from each technique

| SVM  | Yearly (PCA)  |  | Yearly (LLE)  |  |
|------|---|--|---|--|
|      |    |  |    |  |
| SVM  | Career (PCA)  |  | Career (LLE)  |  |
|      |    |  |    |  |
| IQR  | Yearly (Box)  |  | Yearly (Box)  |  |
|      |   |  |   |  |
| IQR  | Career (Box)  |  | Career (Box)  |  |
|      |  |  |  |  |
| GAUS | Yearly (PCA)  |  | Yearly (LLE)  |  |
|      |  |  |  |  |
| GAUS | Career (PCA)  |  | Career (LLE)  |  |
|      |  |  |  |  |



We additionally visualise the low-density regions of the Gaussian Mixture technique in Figure 5

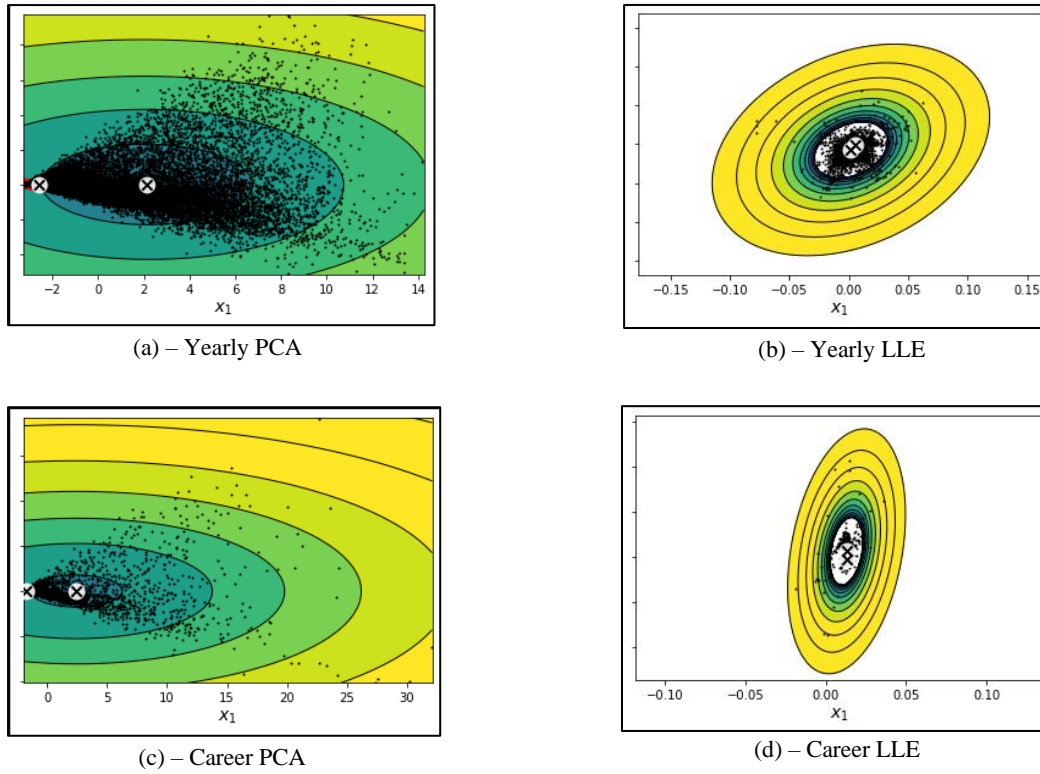


Figure 5: Gaussian Mixture density regions for outlier detection

As mentioned earlier, the task is not simply to perform outlier detection, but rather to detect the outstanding (i.e., good) players. By performing outlier detection, we have also considered the other end of the spectrum (i.e., the bad players). Therefore, we filter these results based on the top “pts” (points) feature which we showed to be a good indicator of player performance in the exploratory data analysis section. A summary of each technique’s top 5 career players using LLE dimensionality reduction is presented in Figure 6.

| ilkid     | gp       | minutes  | pts       | oreb      | dreb      | reb       | asts     | stl       | blk       | turnover  | pf       | fga      | fgm       | fta       | ftm       | tpa       | tpm       |
|-----------|----------|----------|-----------|-----------|-----------|-----------|----------|-----------|-----------|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.729470  | 9.625047  | 9.965913  | 5.762868 | 4.188157  | 14.735923 | 5.302706  | 6.135618 | 9.052888 | 10.896267 | 8.461610  | 7.998601  | -0.220064 | -0.242068 |
| MALONKA01 | 5.121358 | 6.891306 | 9.742362  | 7.877919  | 11.883973 | 8.486746  | 5.311623 | 7.856104  | 5.099507  | 9.832439  | 6.021315 | 8.344009 | 9.235315  | 12.194936 | 11.879100 | 0.530215  | 0.354486  |
| JORDAMI01 | 3.536638 | 5.015434 | 8.457075  | 3.541987  | 4.970613  | 3.525285  | 5.733296 | 9.557238  | 3.911456  | 6.203209  | 3.424169 | 7.778461 | 8.274279  | 7.950248  | 8.774701  | 4.302166  | 3.876996  |
| CHAMBWI01 | 3.430728 | 5.943475 | 8.215044  | -0.373723 | -0.398738 | 13.844166 | 4.648994 | -0.411646 | -0.298583 | -0.429209 | 2.399779 | 7.426895 | 8.626036  | 10.920376 | 7.172023  | -0.266314 | -0.249170 |
| HAYESEL01 | 4.442753 | 6.233666 | 7.076695  | 6.259107  | 7.041740  | 9.271488  | 2.190148 | 3.014414  | 8.050778  | 2.651100  | 5.464267 | 7.688879 | 7.399564  | 7.207236  | 6.287395  | -0.178953 | -0.213661 |

(a) – SVM

| ilkid     | gp       | minutes  | pts      | oreb      | dreb      | reb      | asts     | stl       | blk       | turnover  | pf       | fga      | fgm      | fta      | ftm      | tpa       | tpm       |
|-----------|----------|----------|----------|-----------|-----------|----------|----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|-----------|-----------|
| WESTDOO1  | 1.983297 | 1.713276 | 1.300111 | 0.896495  | 0.815555  | 0.533456 | 0.978797 | 1.408449  | 0.587741  | 1.437576  | 2.091594 | 1.333987 | 1.396686 | 0.931519 | 1.014963 | 0.232159  | 0.013598  |
| MENGEJO01 | 1.826394 | 1.189957 | 1.228305 | 0.110965  | 0.336454  | 0.264897 | 1.019321 | 0.873127  | -0.166577 | 0.205906  | 1.255299 | 1.189980 | 1.243467 | 1.160286 | 1.272401 | -0.250897 | -0.249170 |
| ROBINBI01 | 1.406679 | 1.006165 | 1.040614 | 2.231179  | 1.861256  | 1.453973 | 0.177070 | 1.820845  | 0.097434  | 1.315090  | 2.237729 | 1.016225 | 1.075862 | 1.018989 | 1.003606 | -0.245758 | -0.234966 |
| STEELLA01 | 1.724407 | 1.459546 | 0.893122 | 0.648181  | 0.579739  | 0.599848 | 1.446470 | 2.943038  | 0.154008  | 0.013103  | 2.048188 | 0.896895 | 0.957171 | 0.689295 | 0.751215 | -0.256036 | -0.249170 |
| NEUMAPA01 | 1.100718 | 0.948018 | 0.887577 | -0.373723 | -0.398738 | 0.322915 | 1.155133 | -0.411646 | -0.298583 | -0.429209 | 1.268321 | 0.882022 | 0.818339 | 1.123760 | 1.213089 | -0.266314 | -0.249170 |

(b) – IQR

| ilkid     | gp       | minutes  | pts       | oreb    | dreb     | reb      | asts     | stl      | blk       | turnover | pf       | fga      | fgm       | fta     | ftm      | tpa       | tpm       |
|-----------|----------|----------|-----------|---------|----------|----------|----------|----------|-----------|----------|----------|----------|-----------|---------|----------|-----------|-----------|
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.72947 | 9.625047 | 9.965913 | 5.762868 | 4.188157 | 14.735923 | 5.302706 | 6.135618 | 9.052888 | 10.896267 | 8.46161 | 7.998601 | -0.220064 | -0.242068 |
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.72947 | 9.625047 | 9.965913 | 5.762868 | 4.188157 | 14.735923 | 5.302706 | 6.135618 | 9.052888 | 10.896267 | 8.46161 | 7.998601 | -0.220064 | -0.242068 |
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.72947 | 9.625047 | 9.965913 | 5.762868 | 4.188157 | 14.735923 | 5.302706 | 6.135618 | 9.052888 | 10.896267 | 8.46161 | 7.998601 | -0.220064 | -0.242068 |
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.72947 | 9.625047 | 9.965913 | 5.762868 | 4.188157 | 14.735923 | 5.302706 | 6.135618 | 9.052888 | 10.896267 | 8.46161 | 7.998601 | -0.220064 | -0.242068 |
| ABDULKA01 | 5.450855 | 7.242896 | 10.146856 | 6.72947 | 9.625047 | 9.965913 | 5.762868 | 4.188157 | 14.735923 | 5.302706 | 6.135618 | 9.052888 | 10.896267 | 8.46161 | 7.998601 | -0.220064 | -0.242068 |

(c) – Gaussian Mixture

Figure 6: Top 5 outstanding players from each technique (Career LLE)

We present a detailed discussion of these results in Section 4. However, the following key findings can be noted. Firstly, LLE is superior to PCA for catering towards outlier detection; SVM and Gaussian mutually agree on the outstanding player(s), however SVM seems more sensitive at lower dimensions.

## 3.2 PREDICTING GAME OUTCOMES

The next major task involves trying to predict the outcome of a game when given 2 competing teams. Due to the nature of the data, that is we are not given any match history but rather only a single teams win/loss stats, we do not have any prior history of how these two teams faired in prior engagements. Therefore, I approach this task from a regression perspective, trying to predict the win percentage of each team given their prior stats and returning the team with the higher win percentage as the winner. We repeat this process for multiple regressors to form an ensemble of regressors with majority-voting.

However, before we reach this stage, we follow the machine learning workflow presented in Figure 1. The following machine learning techniques were evaluated at each stage for this task:

- **Data Pre-processing**
  - Feature engineering
  - Min-Max Scaling
  - Hold-out cross validation
- **Exploratory Data Analysis**
  - Box-Plot distributions
  - Correlation heatmaps
  - Win percentage distributions
- **Dimensionality Reduction**
  - Principal Component Analysis (PCA)
  - Local Linear Embedding (LLE)
- **Game outcome predictors (Regression)**
  - Linear regression
  - Polynomial regression
  - Ensemble of regressors

### 3.2.1 Data Pre-processing

We similarly perform data pre-processing as the first phase to ensure our data takes the correct form and all necessary features are accounted for. We first load the “teams” data and remove the unnecessary non-numerical columns, but making sure to keep the team’s name as the primary identifier. We then feature engineer a new column which calculates the win percentage for a given team, in that year. As mentioned earlier, this win percentage figure is what our regression models will be trying to predict and we subsequently choose the team with the higher win percentage as the winner. The next pre-processing technique we perform is Min-Max scaling of the data which allocates each feature value into a [0,1] range which is dependent on the minimum and maximum values seen in the data.

Figure 7 presents a comparison of the original, unprocessed, data with our scaled and feature-engineered dataset.

|   | team | o_fgm | o_fga | o_ftm | o_fta | o_oreb | o_dreb | o_reb | o_ast | o_pf | ... | d_pf | d_stl | d_to | d_blk | d_3pm | d_3pa | d_pts    | pace | won | lost |
|---|------|-------|-------|-------|-------|--------|--------|-------|-------|------|-----|------|-------|------|-------|-------|-------|----------|------|-----|------|
| 0 | BOS  | 1397  | 5133  | 811   | 1375  | 0      | 0      | 0     | 470   | 1202 | ... | 0    | 0     | 0    | 0     | 0     | 3900  | 0.000000 | 22   | 38  |      |
| 1 | CHI  | 1879  | 6309  | 939   | 1550  | 0      | 0      | 0     | 436   | 1473 | ... | 0    | 0     | 0    | 0     | 0     | 4471  | 0.000000 | 39   | 22  |      |
| 2 | CL1  | 1674  | 5699  | 903   | 1428  | 0      | 0      | 0     | 494   | 1246 | ... | 0    | 0     | 0    | 0     | 0     | 4308  | 0.000000 | 30   | 30  |      |
| 3 | DE1  | 1437  | 5843  | 923   | 1494  | 0      | 0      | 0     | 482   | 1351 | ... | 0    | 0     | 0    | 0     | 0     | 3918  | 0.000000 | 20   | 40  |      |
| 4 | NYK  | 1465  | 5255  | 951   | 1438  | 0      | 0      | 0     | 457   | 1218 | ... | 0    | 0     | 0    | 0     | 0     | 3840  | 0.000000 | 33   | 27  |      |

(a) – Before

|   | o_fgm    | o_fga    | o_ftm    | o_fta    | o_oreb   | o_dreb   | o_reb    | o_ast    | o_pf     | o_stl    | ... | d_to     | d_blk    | d_3pm    | d_3pa    | d_pts    | pace     | team | win_pct  | won | lost |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----|----------|----------|----------|----------|----------|----------|------|----------|-----|------|
| 0 | 0.266060 | 0.497039 | 0.248536 | 0.347592 | 0.000000 | 0.000000 | 0.000000 | 0.108051 | 0.428056 | 0.000000 | ... | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.288159 | 0.000000 | BOS  | 0.366667 | 22  | 38   |
| 1 | 0.398952 | 0.639154 | 0.302092 | 0.403042 | 0.000000 | 0.000000 | 0.000000 | 0.093644 | 0.550293 | 0.000000 | ... | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.347731 | 0.000000 | CHI  | 0.639344 | 39  | 22   |
| 2 | 0.342432 | 0.565438 | 0.287029 | 0.364385 | 0.000000 | 0.000000 | 0.000000 | 0.118220 | 0.447903 | 0.000000 | ... | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.330725 | 0.000000 | CL1  | 0.500000 | 30  | 30   |
| 3 | 0.277089 | 0.582840 | 0.295397 | 0.385298 | 0.000000 | 0.000000 | 0.000000 | 0.113136 | 0.495264 | 0.000000 | ... | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.290037 | 0.000000 | DE1  | 0.333333 | 20  | 40   |
| 4 | 0.284808 | 0.511782 | 0.307113 | 0.367554 | 0.000000 | 0.000000 | 0.000000 | 0.102542 | 0.435273 | 0.000000 | ... | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.281899 | 0.000000 | NYK  | 0.550000 | 33  | 27   |

(b) – After

Figure 7: Data pre-processing output comparison

A key step taken at this phase is the generation of the training, validation, and testing sets for our models. The pre-processed team’s dataset is separated into the training and validation set, via an 80/20 respective split. As we shown in the next section, this size is congruent with other similar splitting procedures as it reaches the minimum Root Mean Squared Error (RMSE). Finally, to generate our testing set, we permute through every single possible combination of team matches. This results in an overwhelming amount of ‘9120’ possible matches. We subsequently reduce our testing set to ‘100’ matches for which our regressors will have to predict the win rate for each team in the match.

### 3.2.2 Data Analysis

We once again perform some data analysis to see if we can, manually, gather some key insights from our pre-processed data. To do this, we similarly evaluate the distribution and correlations of the features using the box-plots and correlation heatmaps respectively. Additionally, we visualize the distribution of our focal win percentages which we feature engineered in the previous section. The resulting visualizations for each of these techniques are presented in Figure 8.

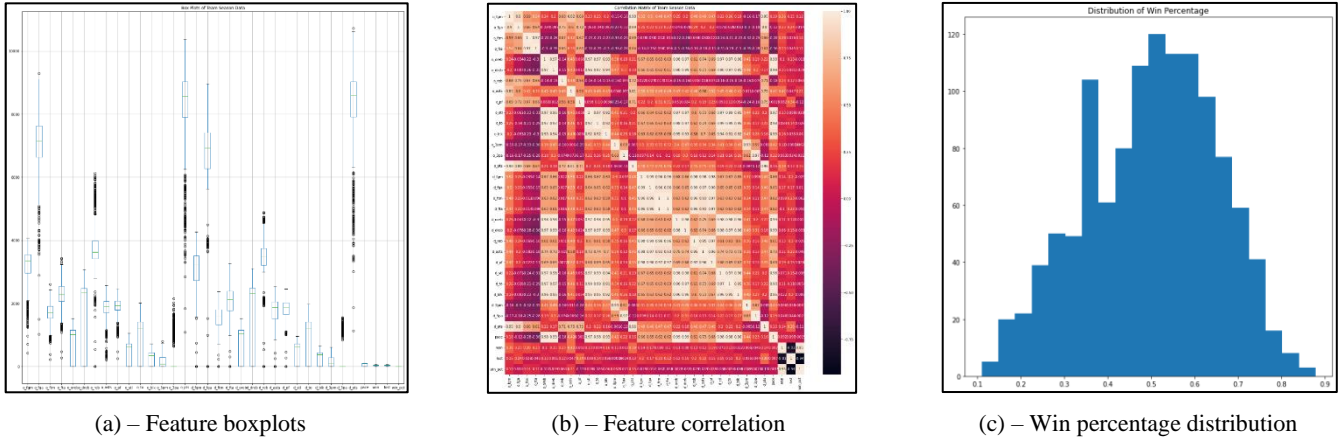


Figure 8: Data Analysis Plots

In particular, we can see that the win percentages follow a somewhat normal distribution. This is to be expected as there can only be one winner for each game so there will subsequently be an equal number of wins as there are losses. This does however allude to the notion of ‘more confident’ predictions, as a team with a relatively higher win-rate is more likely to win a game against a team on the opposite end of the spectrum.

### 3.2.3 Dimensionality Reduction

Dimensionality reduction is particularly important for this task as we have over 30 features to consider. However, as we have shown, with exploratory data analysis, some of these features are not required. Hence, we can use dimensionality reduction techniques to reduce our data to a more computationally feasible, and visually representative feature space. We evaluate both a projection and manifold learning approach to dimensionality reduction with PCA and LLE respectively. When evaluating potential feature spaces, we allowed for PCA to project down to the number of dimensions which preserved 95% of the original variance. By doing this, we found that a 3D feature space preserved this desired variance, and subsequently used LLE with 3 components to create a mutual 3D feature space. The resulting dimensionality reduction outputs can be seen in Figure 9.

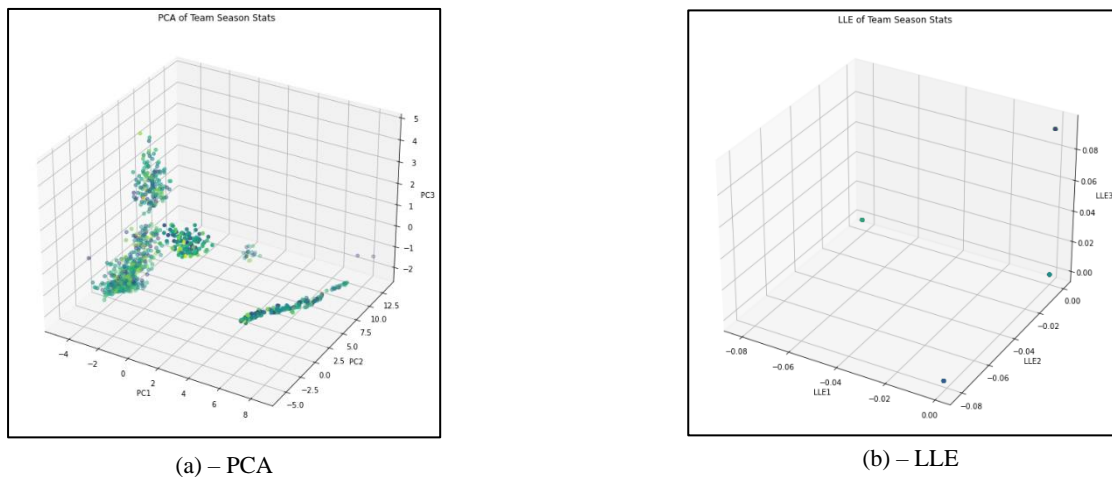


Figure 9: Dimensionality reduction technique outputs

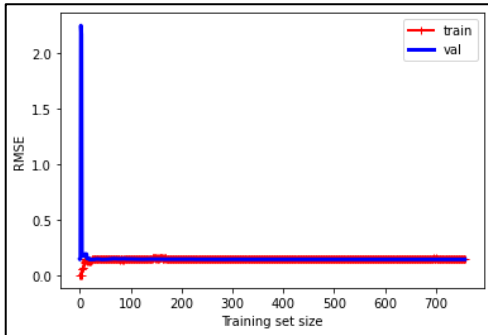
We once again take the same stance we did in the outlier detection task, of using the results of both techniques. I.e., we carry forward PCA and LLE to the regressors to determine what the joint effect of each technique has on the overall predictions.

### 3.2.4 Game Predictions

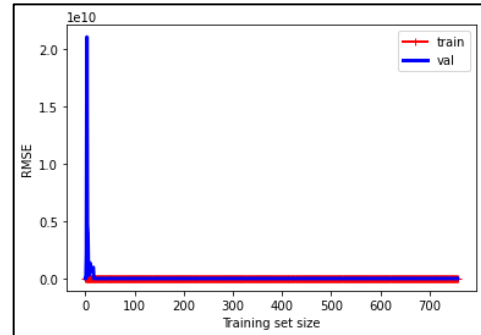
After we have completed the aforementioned stages, we are ready to make predictions on the ‘100’ randomly selected team combinations. Since this is modelled as a regression task, we compare two popular regression methods: Linear Regression and Polynomial Regression. Each regressor is trying to predict the win rate for each team. This is done by averaging the historic match data for each team and using this average as the input vector for our regressors. We then compare the predicted win percentage for each team and nominate the team with the higher win percentage the victor for the match.

#### 3.2.4.1 Linear Regression

A linear regressor is a linear model that makes a prediction by simply computing the weighted sum of the input vector, plus a constant bias term [1]. We use the Root Mean Square Error (RSME) as a measure to find out how well, or poorly, the linear regressor fits to the training data. We visualise the trend that the training data has on the RMSE for both dimensionality reduction techniques. This is shown in Figure 10.



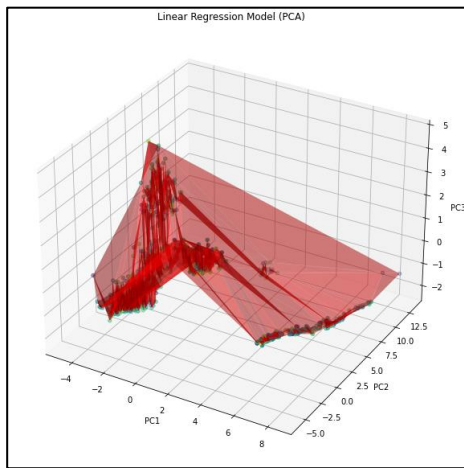
(a) – Linear regression learning curve (PCA)



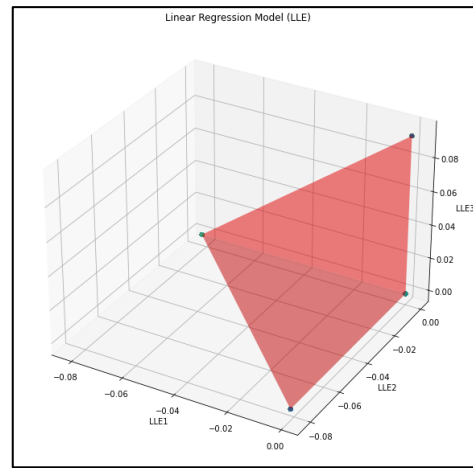
(b) – Linear regression learning curve (LLE)

Figure 10: RMSE learning curves

Once the training data has been fit to the linear regressor, we visualise the resulting curve upon which the input vectors will be ‘projected’ onto. We do this once again for both the PCA and LLE-based reduced training data. This is shown in Figure 11.



(a) – Linear regression model (PCA)



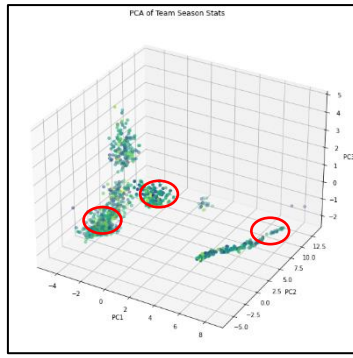
(b) – Linear regression model (LLE)

Figure 11: Resulting linear regression models

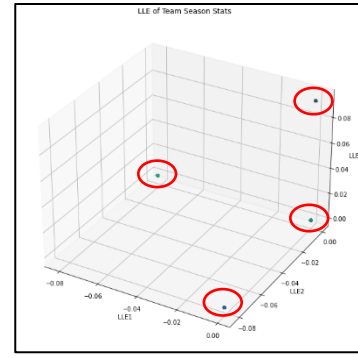
We finally make the predictions for both teams and output the team with the higher win percentage as the predicted winner of the game. For a performance measure, we take note of the mean squared error and R-2 scores, described in Table 1. This process of making each team’s predictions the predictions, calculating the metrics and predicting the final winner is shown in the below figures and table.

Table 4: Metrics gathered

| MODEL        | MSE  | R2 SCORE |
|--------------|------|----------|
| Linear (PCA) | 0.02 | -0.03    |
| Linear (LLE) | 0.02 | -0.01    |



(a) – Linear predictions (PCA)



(b) – Linear predictions (LLE)

Figure 12: Predictions for each team plotted on the resulting models

| team1 | team2 | Linear PCA Winner | Linear LLE Winner |
|-------|-------|-------------------|-------------------|
| 8225  | TRI   | NYK               | TRI               |
| 3133  | INJ   | WAT               | WAT               |
| 1446  | CLT   | DLC               | DLC               |
| 4568  | MNM   | CAR               | CAR               |
| 6022  | PIT   | LAS               | LAS               |
| ...   | ...   | ...               | ...               |
| 4830  | NJA   | STL               | NJA               |
| 2303  | FLA   | DN1               | FLA               |
| 2867  | IND   | DAL               | IND               |
| 4209  | MIN   | IND               | MIN               |
| 5908  | PHO   | DE1               | PHO               |

100 rows × 4 columns

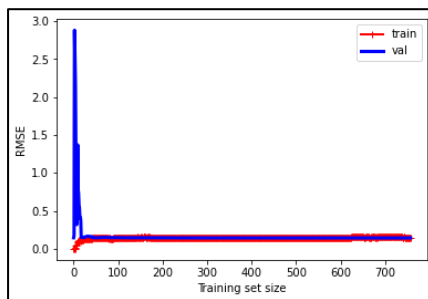
Figure 13: Final predictions for the winner of the game

A more detailed discussion of these results is presented in Section 4.2; however, we can briefly take note of the following main observations. At this stage the only aspect to compare is the different dimensionality reduction techniques. Using the metrics, both obtain the same MSE, but surprisingly LLE achieves a better R2 score, this indicates that the independent variables are accounting for more variation for the dependant than PCA. However, these values are very close, so one might argue this is within margin of error.

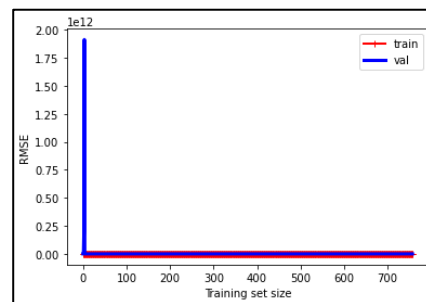
### 3.2.4.2 Polynomial Regression

Polynomial regression is used in the scenarios where the data takes a much more complex form than a straight line. It fits a linear model to the non-linear data by adding new features with a specified polynomial degree. The polynomial regressor will add new features to the already reduced feature space of the training set. Therefore, mostly to make the feature space visible once again, we perform dimensionality reduction on top of this scaling before eventually fitting the linear model.

Once this process is concluded, we follow a similar methodology to that described for the linear regressor. We first plot the RMSE curves to determine the quality of the fit to the training data. This is presented in Figure 14.



(a) – Polynomial regression learning curve (PCA)



(b) – Polynomial regression learning curve (LLE)

Figure 14: RMSE learning curves



We then visualise the resulting polynomial regressors in the 3-dimensional feature space for both dimensionality reduction techniques. This is presented in Figure 15.

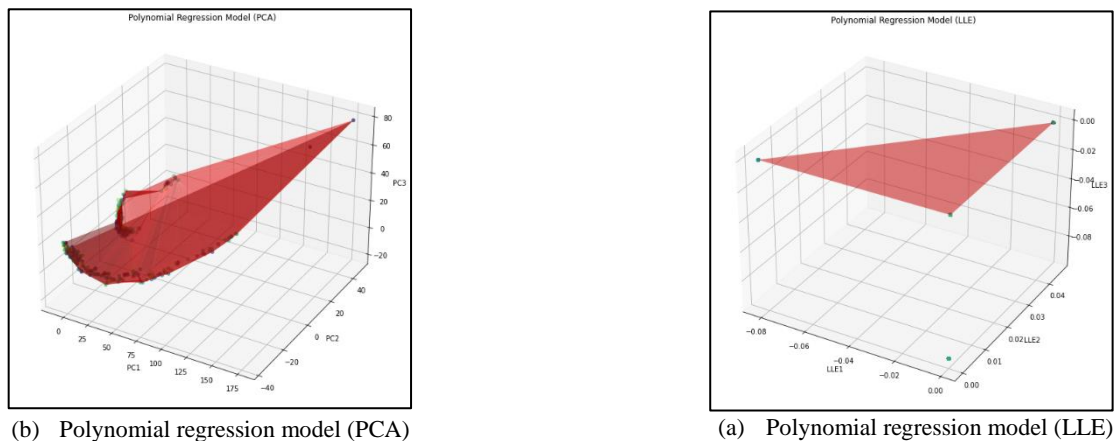


Figure 15: Resulting polynomial regression models

Lastly, we use the resulting polynomial regressor to make a prediction for each team's win percentage. Following this, we plot the results onto the regressor and take note of the metrics. Finally, we compare the resulting predictions made for each team and predict the team with the higher win percentage as the game winner. The results of this process are presented in the figures and table below.

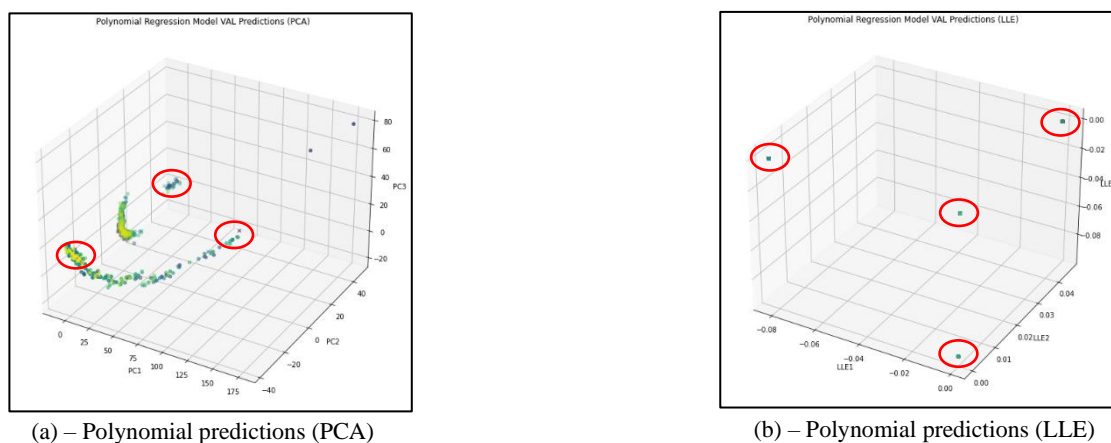


Figure 16: Predictions for each team plotted on the resulting models

Table 5: Metrics

| MODEL            | MSE  | R2 SCORE |
|------------------|------|----------|
| Polynomial (PCA) | 0.02 | -0.01    |
| Polynomial (LLE) | 0.02 | -0.01    |

| team1 | team2 | Linear PCA Winner | Linear LLE Winner | Polynomial PCA Winner | Polynomial LLE Winner |
|-------|-------|-------------------|-------------------|-----------------------|-----------------------|
| 8225  | TRI   | NYK               | TRI               | NYK                   | NYK                   |
| 3133  | INJ   | WAT               | WAT               | WAT                   | INJ                   |
| 1446  | CL1   | DLC               | DLC               | DLC                   | CL1                   |
| 4568  | MNM   | CAR               | CAR               | CAR                   | MNM                   |
| 6022  | PIT   | LAS               | LAS               | LAS                   | PIT                   |
| —     | —     | —                 | —                 | —                     | —                     |
| 4830  | NJA   | STL               | NJA               | NJA                   | STL                   |
| 2303  | FLA   | DN1               | FLA               | FLA                   | DN1                   |
| 2867  | IND   | DAL               | IND               | DAL                   | DAL                   |
| 4209  | MIN   | IND               | MIN               | IND                   | IND                   |
| 5908  | PHO   | DE1               | PHO               | PHO                   | DE1                   |

Figure 17: Final predictions for the winner of the game

A more detailed discussion of these results is presented in Section 4.2; however, we can make a similar observation to that made in Section 3.2.4.1, and this time the metrics produce identical results even with vastly different approaches.

### 3.2.4.3 Regression Ensemble

The last technique involves creating an ensemble of the four regressor models generated in the previous sections. This is done to imitate a more realistic scenario of making a prediction for a game. Rather than asking a single person to give a prediction for a game, it is better to ask around and get multiple peoples' views on which team they think will win. More often than not, the aggregated predictions of the general public will outperform getting a single prediction, even if that prediction comes from an expert. This methodology is embodied in the ensemble machine learning technique.

We aggregate the results of the predictions made by each of the 4 regressor models, and use a hard-voting (majority vote) for the winner of the game. The architecture of this predictor is given in Figure 18.

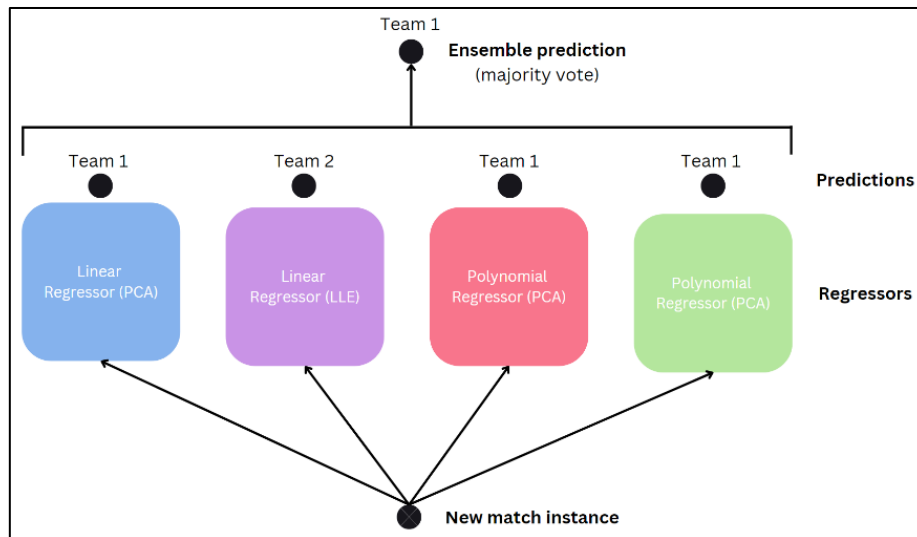


Figure 18: Ensemble architecture using previous regressors

The resulting aggregated predictions resulting from this architecture is presented in Figure 19.

|      | team1 | team2 | Linear PCA Winner | Linear LLE Winner | Polynomial PCA Winner | Polynomial LLE Winner | Winner |
|------|-------|-------|-------------------|-------------------|-----------------------|-----------------------|--------|
| 8225 | TRI   | NYK   | NYK               | TRI               | NYK                   | NYK                   | NYK    |
| 3133 | INJ   | WAT   | WAT               | WAT               | WAT                   | INJ                   | WAT    |
| 1446 | CL1   | DLC   | DLC               | DLC               | DLC                   | CL1                   | DLC    |
| 4568 | MNM   | CAR   | CAR               | CAR               | CAR                   | MNM                   | CAR    |
| 6022 | PIT   | LAS   | LAS               | LAS               | LAS                   | PIT                   | LAS    |
| ...  | ...   | ...   | ...               | ...               | ...                   | ...                   | ...    |
| 4830 | NJA   | STL   | NJA               | NJA               | NJA                   | STL                   | NJA    |
| 2303 | FLA   | DN1   | FLA               | FLA               | FLA                   | DN1                   | FLA    |
| 2867 | IND   | DAL   | IND               | IND               | DAL                   | DAL                   | DAL    |
| 4209 | MIN   | IND   | IND               | MIN               | IND                   | IND                   | IND    |
| 5908 | PHO   | DE1   | PHO               | PHO               | PHO                   | DE1                   | PHO    |

Figure 19: Aggregated ensemble predictions for game winner

If we then follow the notion of “the wisdom of the crowd” and treat the ensembled predictions as the true outcome of the game, then we can create a classifier-esque scenario and compare the various models. The resulting confusion matrices from this simulated classification task is presented in Figure 21, and a summary of the classification reports is presented in Table 6, using the macro averages where possible.

Table 6: Summary of classification metrics

|                  |                | Metric   |          |           |        |
|------------------|----------------|----------|----------|-----------|--------|
|                  |                | Accuracy | F1-Score | Precision | Recall |
| Regression Model | Linear PCA     | 0.76     | 0.76     | 0.76      | 0.76   |
|                  | Linear LLE     | 0.53     | 0.53     | 0.53      | 0.53   |
|                  | Polynomial PCA | 0.84     | 0.84     | 0.84      | 0.84   |
|                  | Polynomial LLE | 0.54     | 0.54     | 0.54      | 0.54   |

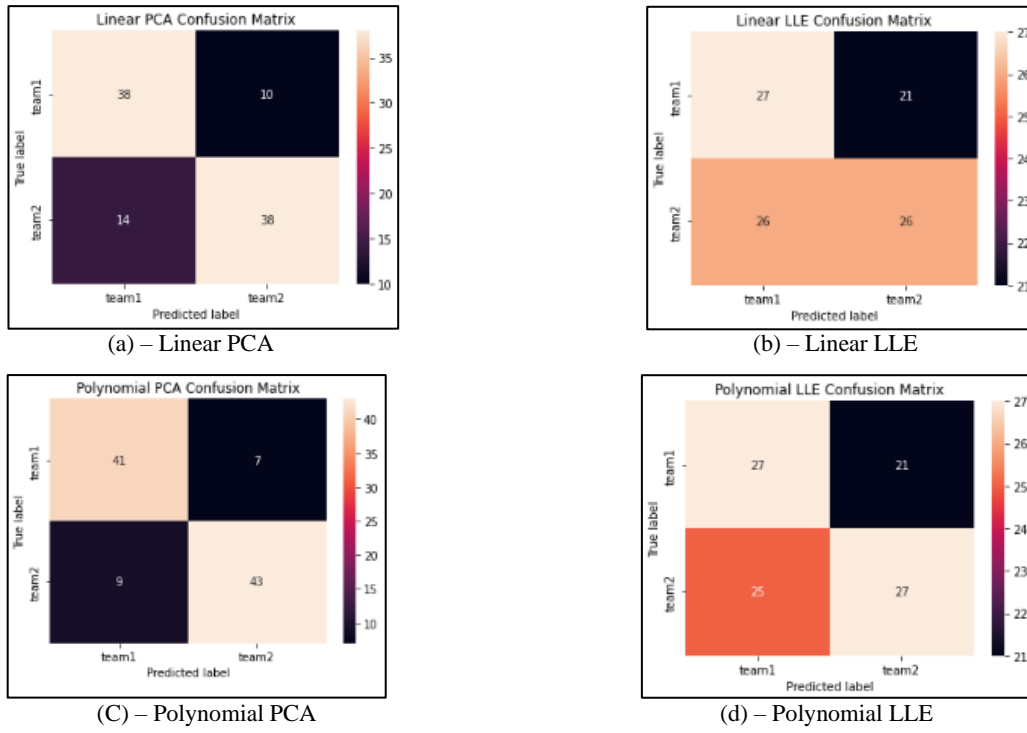


Figure 20: Confusion matrices for team outcome predictions

A more detailed discussion is presented in Section 4.2; however, we can note that PCA outperforms LLE in this task, and that between the regression classifiers themselves, the polynomial regression performed the best.

## 4 RESULTS AND DISCUSSION

In wake of the results and discussions already made in Sections 3.1 – 3.2, a summary of their findings is presented in this section, along with a more detailed analysis of the outcomes for each task. I.e., since the results have been presented in Section 3, this section mostly presents a more in-depth discussion of the results for each task.

### 4.1 OUTSTANDING PLAYERS (OUTLIER DETECTION)

For the evaluated machine learning methods, in this task, we were able to come to the following main conclusions, based on the isolated and joint results presented in Section 3.1:

- Isolated result: Local Linear Embedding was the best dimensionality reduction technique
- Isolated result: Gaussian Mixture was the best performing outlier detector
- Joint result: When paired with LLE, the Gaussian Mixture was best able to detect outstanding players across both data set (yearly and career player stats)

In reaching these conclusions it with worth touching on the prior machine learning techniques which ultimately lead to the effective detection of these players. The most notable pre-processing technique was the use of standard scaling to allow our features to follow the mean with a unit standard deviation. Whilst min-max scaling might seem to the best replacement, it was needed to enforce the SVM hinge loss. Furthermore, through the various analysis techniques we showed that there these outstanding players still remained after scaling.

Moving onto the main conclusions, we showed that the LLE dimensionality reduction technique was superior to PCA. This could be visually noted by the distribution of the players in the lower dimensional feature space (see Table 2). We can notice that LLE creates a clearer central clustering of the inlier datapoints with outliers being clearly distinguishable. In contrast PCA's cone-like spread makes it hard to detect the inliers. To explain this difference, we can refer to the underling technique of dimensionality reduction that each of these techniques use. PCA is a form of projection dimensionality reduction, whereas LLE is a manifold learning technique. Projection methods make the assumption that the training instances lie within, or close to, a much lower-dimensional subspace of the higher-dimensional space [1].



However manifold learning is capable of tackling scenarios where the data cannot be projected on this lower-dimensional space without risking an overlap of training instances. Rather, for the case of 2D dimensionality reduction, it will create a curved hyperplane that ‘rolls’ in the third-dimension but resembles a 2D plane upon which the training instances are projected. We can see that manifold learning is the preferred technique as there is overlapping of training instances in the projection-based PCA dimensionality reduction technique.

The next main result that was noted from this task was that the Gaussian Mixture (GM) anomaly detection technique outperformed the SVM and IQR techniques. We were first able to eliminate the IQR evaluation of outstanding players due to its outputs not agreeing with the other techniques, and the players themselves having objectively worse statistics than the other techniques’ outstanding players. The IQR is the most intuitive way to identify outliers from a statistical background however it seems that when in a significantly reduced feature space, the upper and lower thresholds it identifies for each feature is insignificant. When comparing the SVM and the GM techniques, we can see that the GM generates larger inlier clusters than the SVM, particularly for the yearly player stats using LLE (see Table 3). This allows the GM to be more stringent on its identification of outliers. Furthermore, from Figure 5, we can note that the outer density regions, of the LLE-reduced space, is clearly able to detect any outliers whilst clustering the inliers as the high-density regions. Whilst the SVM can be used for outlier detection, it is mostly used for classification tasks. Congruently, we can see this novelty in lower-dimensional spaces as it becomes more sensitive to outliers, which we have shown we have shown.

Also, as a quick sanity check, when we look at the most outstanding player detected from our best combination of machine learning techniques (standard scaling + LLE + GM), we find that Kareem Abdul-Jabbar is the most outstanding player. By doing external research, we can confirm this result, by observing that Kareem is a highly rated player and considered one of the best, and we have further substantiated this view based on machine learning techniques on his player stats relative to the competition.

## 4.2 GAME PREDICTIONS

For the evaluated machine learning methods, in this task, we were able to come to the following main conclusions, based on the isolated and joint results presented in Section 3.2:

- Isolated result: Principal Component Analysis is the best evaluated dimensionality reduction technique
- Isolated result: Polynomial regression is the best single predictor
- Joint result: Ensemble of regressors (both polynomial and linear) is the best overall predictor

Note, since the MSE and R2-Score were not able to discriminate between the dimensionality reduction techniques, we note the above observations from the ensemble classification.

Firstly, we observe that PCA is the best reduction technique for this task. This is in contrast to task 1 where we found that LLE was best for producing a favourable distribution of training instances for outlier detection. By looking at figure 9 we can make the visual observation that PCA allows for a better discrimination of points, whereas LLE has 4 main anchoring points which forms a 2D-like hyperplane upon which the predictions are made, effectively indicating that there is a redundant component. Analysing this, we refer the reader Section 4.1’s discussion about PCA being a project technique whereas LLE is a manifold technique.

Secondly, the polynomial regressor was shown to be the best regression technique as it obtained the highest accuracy, f1-score, precision and recall amongst all evaluated regressors. When analysing Figure 9 and Figure 15, we can see that the data is non-linear, this inherently makes the polynomial more favourable than the linear regressor as the polynomial regressor is capable of fitting to this data more precisely. It does this by fitting a linear model to the non-linear data by adding new features with a specified polynomial degree. Resultantly, this non-linear curve allows for closer, and more accurate, predictions than a linear alternative.

Finally, by combining all 4 regressors into a majority voting ensemble, we are able to leverage “the wisdom of the crowd” paradigm which states that having numerous and diverse models will often outperform a strong learner. In the context of sporting predictions, this makes sense, as you would most likely ask numerous people who they think would win than just a single person, even if he/she happens to be an expert. This diverse set of regressors, both linear and polynomial, and PCA-reduced and LLE-reduced allows each other to complement their individual weaknesses, and therefore present a robust and, generally, more accurate prediction.

## 5 CONCLUSION

---

In this research, we have performed an empirical evaluation of various machine learning techniques across two tasks: (1) Outstanding player detection, and (2) Game predictions. As a whole, we evaluated 4 data selection and pre-processing techniques (standard scaling, min-max scaling, feature engineering, and hold-out cross validation), 3 exploratory data analysis methods (boxplots, correlation maps, and distributions), 2 dimensionality reduction techniques (PCA and LLE), 3 outlier detectors (SVM, IQR, and Gaussian mixture), and 3 regression models (linear regression, polynomial regression, and a regression ensemble). This research shows that, for outlier detection, the joint use of standard scaling, Local Linear Embedding and Gaussian mixtures produces the best detection of outstanding players, which was facilitated via an outlier detection operation. For the task of predicting a game outcome, we have shown that the joint use of min-max scaling, principal component analysis, and a regression ensemble produced the best theoretical game predictions, with PCA-based polynomial regression being the strongest model within the ensemble.

The full system source code is available via this [GitHub link](https://github.com/MagneticPanda/NBA-Game-Prediction-and-Outlier-Detection) (<https://github.com/MagneticPanda/NBA-Game-Prediction-and-Outlier-Detection>)

# REFERENCES

- [1] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems, Second edition. Beijing [China] ; Sebastopol, CA: O'Reilly Media, Inc, 2019.
- [2] E. Dixon, 'NBA reveals record US\$10bn revenue for 2021/22', SportsPro, Jul. 15, 2022. <https://www.sportspromedia.com/news/nba-revenue-2021-22-season-adam-silver/> (accessed Nov. 08, 2022).
- [3] S. H. Mohamed, T. E. H. El-Gorashi, and J. M. H. Elmirghani, 'A Survey of Big Data Machine Learning Applications Optimization in Cloud Data Centers and Networks'. arXiv, Oct. 01, 2019. doi: 10.48550/arXiv.1910.00731.
- [4] A. Lalwani, A. Saraiya, A. Singh, A. Jain, and T. Dash, 'Machine Learning in Sports: A Case Study on Using Explainable Models for Predicting Outcomes of Volleyball Matches'. arXiv, Jun. 18, 2022. doi: 10.48550/arXiv.2206.09258.
- [5] C. Li, S. Kampakis, and P. Treleaven, 'Machine Learning Modeling to Evaluate the Value of Football Players'. arXiv, Jul. 22, 2022. doi: 10.48550/arXiv.2207.11361.
- [6] R. Bunker and T. Susnjak, 'The Application of Machine Learning Techniques for Predicting Results in Team Sport: A Review', J. Artif. Intell. Res., vol. 73, pp. 1285–1322, Apr. 2022, doi: 10.1613/jair.1.13509.
- [7] K. Pelechris, 'Implicit Biases in Refereeing: Lessons from NBA Referees'. arXiv, Oct. 24, 2022. doi: 10.48550/arXiv.2210.13687.
- [8] Z. Ivankovic, M. Racković, B. Markoski, D. Radosav, and M. Ivkovic, 'Analysis of basketball games using neural networks', Dec. 2010, pp. 251–256. doi: 10.1109/CINTI.2010.5672237.
- [9] M. Migliorati, 'Features selection in NBA outcome prediction through Deep Learning'. arXiv, Nov. 17, 2021. doi: 10.48550/arXiv.2111.09695.
- [10] I. Bhandari, E. Colet, J. Parker, Z. Pines, R. Pratap, and K. Ramanujam, 'Advanced Scout: Data Mining and Knowledge Discovery in NBA Data', Data Min. Knowl. Discov., vol. 1, no. 1, pp. 121–125, Mar. 1997, doi: 10.1023/A:1009782106822.
- [11] K. Nguyen and S. Chawla, 'Robust Outlier Detection Using Commute Time and Eigenspace Embedding', Jun. 2010, pp. 422–434. doi: 10.1007/978-3-642-13672-6\_41.
- [12] J. Wang and Q. Fan, 'Application of Machine Learning on NBA Data Sets', J. Phys. Conf. Ser., vol. 1802, no. 3, p. 032036, Mar. 2021, doi: 10.1088/1742-6596/1802/3/032036.
- [13] F. Thabtah, L. Zhang, and N. Abdelhamid, 'NBA Game Result Prediction Using Feature Analysis and Machine Learning', Ann. Data Sci., vol. 6, no. 1, pp. 103–116, Mar. 2019, doi: 10.1007/s40745-018-00189-x.