Premier League Season Point Prediction

Group 14: Thomas Blackburn & Kyle Levesque

OPIM 5603
Team Introduction:

This project has been completed by group 14 of the Fall 2019 OPIM 5603 class, Thomas Blackburn and Kyle Levesque. Thomas Blackburn is currently pursuing his MBA with a concentration in Business Analytics. Kyle Levesque is currently pursuing his MBA with a concentration in Business Ethics and Compliance.

Background Information:

Premier League Background Information
The Premier League, formerly known as the English Premier League, is the top level of the English football league system. The top 20 teams within the English football league system compete for the Premier League title every season. Each team plays 38 games a season, receiving the following number of points per match:

- Each team receives 3 points for a win
- Each team receives 1 points for a draw
- Each team receives 0 points for a loss

The team with the most points at the end of the season wins the Premier League title.

Dataset & Model Background Information
The dataset used for this project contains the following statistical categories as inputs for each team during the 2018-2019 Premier League Season:

- Goals per game
- Assists per game
- Shots per game
- Shots on Target per game
- Completed Passes per game
- Attempted Passes per game
- Passing Completion %
- Goals Against per game

The regression model that will be generated will be using the statistical categories above as inputs, and will have an output of Points accumulated by the team during the season (from wins, losses, and draws).

Problem Statement:

Generate a regression model based on statistical categories for each team collected during a season to predict the number of points accumulated by each team at the end of the season.
Once the linear regression model has been generated, we will be using it to try and predict the number of points accumulated by Premier League teams during a different season. We will use the prior seasons data (the 2017-2018 season) and 2017-2018 season point totals to measure the accuracy of the regression model that has been created.

**Data Review & Exploration:**

**Premier League 2018-2019 Dataset**

The table below is a snapshot of the dataset that will be used:

<table>
<thead>
<tr>
<th>Squad</th>
<th>Goals per game</th>
<th>Assists per game</th>
<th>Shots per game</th>
<th>Shots on Target per game</th>
<th>Completed Passes per game</th>
<th>Attempted Passes per game</th>
<th>Passing Completion %</th>
<th>Goals Against per game</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arsenal</td>
<td>1.82</td>
<td>1.37</td>
<td>12.37</td>
<td>4.53</td>
<td>481.8157895</td>
<td>585.2105263</td>
<td>82.3</td>
<td>1.342105263</td>
<td>70</td>
</tr>
<tr>
<td>Bournemouth</td>
<td>1.45</td>
<td>1.13</td>
<td>11.76</td>
<td>4.26</td>
<td>354.1842105</td>
<td>458.7631579</td>
<td>77.2</td>
<td>1.842105263</td>
<td>45</td>
</tr>
<tr>
<td>Brighton</td>
<td>0.92</td>
<td>0.63</td>
<td>9.79</td>
<td>2.87</td>
<td>304.6578947</td>
<td>413.3157895</td>
<td>73.7</td>
<td>1.578947368</td>
<td>36</td>
</tr>
<tr>
<td>Burnley</td>
<td>1.13</td>
<td>0.84</td>
<td>9.45</td>
<td>3.05</td>
<td>266.9736842</td>
<td>385.4736842</td>
<td>69.3</td>
<td>1.789473684</td>
<td>40</td>
</tr>
<tr>
<td>Cardiff City</td>
<td>0.87</td>
<td>0.53</td>
<td>11.03</td>
<td>3.34</td>
<td>190.6052632</td>
<td>309.0789474</td>
<td>61.7</td>
<td>1.815789474</td>
<td>34</td>
</tr>
<tr>
<td>Chelsea</td>
<td>1.61</td>
<td>1.37</td>
<td>16</td>
<td>5.21</td>
<td>600.0789474</td>
<td>697.2894737</td>
<td>86.1</td>
<td>1.026315789</td>
<td>72</td>
</tr>
<tr>
<td>Crystal Palace</td>
<td>1.26</td>
<td>0.87</td>
<td>12.95</td>
<td>3.89</td>
<td>334.9736842</td>
<td>439.0263158</td>
<td>76.3</td>
<td>1.394736842</td>
<td>49</td>
</tr>
<tr>
<td>Everton</td>
<td>1.39</td>
<td>0.89</td>
<td>13.05</td>
<td>4.5</td>
<td>365.2105263</td>
<td>480.5263158</td>
<td>76</td>
<td>1.210526316</td>
<td>54</td>
</tr>
<tr>
<td>Fulham</td>
<td>0.87</td>
<td>0.63</td>
<td>11.92</td>
<td>3.87</td>
<td>398.5526316</td>
<td>506.3157895</td>
<td>78.7</td>
<td>2.131578947</td>
<td>26</td>
</tr>
<tr>
<td>Huddersfield</td>
<td>0.53</td>
<td>0.34</td>
<td>10.55</td>
<td>3.16</td>
<td>342.1052632</td>
<td>461.3947368</td>
<td>74.1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Leicester City</td>
<td>1.26</td>
<td>0.89</td>
<td>13.63</td>
<td>4.89</td>
<td>387.7368421</td>
<td>500.7368421</td>
<td>77.4</td>
<td>1.263157895</td>
<td>52</td>
</tr>
<tr>
<td>Liverpool</td>
<td>2.26</td>
<td>1.45</td>
<td>15.13</td>
<td>6.03</td>
<td>664.5789474</td>
<td>83.5</td>
<td>0.578947368</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Manchester City</td>
<td>2.39</td>
<td>1.87</td>
<td>17.97</td>
<td>6.89</td>
<td>647.4473684</td>
<td>740.1052632</td>
<td>87.5</td>
<td>0.605263158</td>
<td>98</td>
</tr>
<tr>
<td>Manchester Utd</td>
<td>1.71</td>
<td>1.05</td>
<td>13.92</td>
<td>5.95</td>
<td>441.7894737</td>
<td>544.8157895</td>
<td>81.1</td>
<td>1.421052632</td>
<td>66</td>
</tr>
<tr>
<td>Newcastle Utd</td>
<td>1.08</td>
<td>0.87</td>
<td>11.74</td>
<td>3.76</td>
<td>285.3157895</td>
<td>398.6315789</td>
<td>71.6</td>
<td>1.263157895</td>
<td>45</td>
</tr>
<tr>
<td>Southampton</td>
<td>1.16</td>
<td>0.71</td>
<td>12.79</td>
<td>4.29</td>
<td>304.4210526</td>
<td>421</td>
<td>72.3</td>
<td>1.710526316</td>
<td>39</td>
</tr>
<tr>
<td>Tottenham</td>
<td>1.74</td>
<td>1.24</td>
<td>14.13</td>
<td>5.0</td>
<td>491.6578947</td>
<td>597.4210526</td>
<td>82.3</td>
<td>1.026315789</td>
<td>71</td>
</tr>
<tr>
<td>Watford</td>
<td>1.34</td>
<td>1.03</td>
<td>11.42</td>
<td>4.03</td>
<td>334.5842105</td>
<td>447.4736842</td>
<td>74.8</td>
<td>1.552631579</td>
<td>50</td>
</tr>
<tr>
<td>West Ham</td>
<td>1.34</td>
<td>0.87</td>
<td>11.76</td>
<td>4.21</td>
<td>365.4736842</td>
<td>472.8421053</td>
<td>77.3</td>
<td>1.447368421</td>
<td>52</td>
</tr>
<tr>
<td>Wolves</td>
<td>1.21</td>
<td>0.92</td>
<td>12.53</td>
<td>3.97</td>
<td>368.7631579</td>
<td>474.6052632</td>
<td>77.7</td>
<td>1.210526316</td>
<td>57</td>
</tr>
</tbody>
</table>

As mentioned above, the following statistics have been measured for each team throughout the course of the season and will be used to generate the regression model:

- Goals per game
- Assists per game
- Shots per game
- Shots on Target per game
- Completed Passes per game
- Attempted Passes per game
- Passing Completion %
- Goals Against per game
Data Exploration
There is fairly strong correlation between all of these variables except for Goals Against per game. Goals Against per game has an inverse correlation to the rest of the variables because better performing teams generally excel at all the other metrics and allow less goals; and vice versa.

Methodology:
Based on the desired output of Points accumulated throughout the season, we chose to use a linear regression model for this dataset. The following steps will be performed in the creation of the linear regression model:

- Transformations (If needed)
- Checking 5 assumptions
  - Global Stat
  - Skewness
  - Kurtosis
  - Link Function
  - Heteroscedasticity
- Checking/removing collinearity
- Regression
- Accuracy Analysis
Linear Regression Model Creation:

First Model Creation

The first linear regression model was created using all available input variables without any transformations. The first model generated had a high adjusted R-squared value (above 0.85) & low p-value, so we can continue with this model to see if all 5 assumptions are satisfied.

Checking 5 Assumptions

All 5 assumptions must be acceptable using the gvlma function before continuing. In this case, the Link Function assumption was not satisfied so this model cannot be used.
Transformation & Second Model Creation

Because not all 5 conditions were acceptable, a transformation is required. We transformed the dataset in the following way to create a second model:

- 3 variables have been transformed via a quadratic model
- 3 variables have been transformed via a reciprocal model

Like the first model, the second model generated also had a high adjusted R-squared value (above 0.85) & low p-value, so we could continue with this model to see if all 5 assumptions were satisfied.
Checking 5 Assumptions & Residuals – Model 2

After transformation, the second model needed to be checked to see if all assumptions were acceptable. In this case, all 5 assumptions are acceptable, so we could continue with this linear regression model. Because assumptions were met, we checked residuals plot as well to make sure it is a good fit of the dataset.
Checking & Removing Collinearity – Model 2
Now that we had a working model that satisfies all 5 conditions, we checked for collinearity between variables using VIF function. If a VIF value is greater than 5 for at least one variable, we eliminated the variable with the highest VIF, and then iteratively rerun the VIF function. Continue until there is no VIF value greater than 5.

The first run of VIF function showed VIF values greater than 5, so needed to do the iterative process to remove largest VIF value per iteration. The images of the code below detail the...
process of how the largest VIF value was removed during each iteration until there were no VIF values greater than 5.

```r
#VIF Check - Model 2
```

```r
library(car)

# If VIF > 5 for at least one variable, eliminate the variable with the highest VIF, and then iteratively rerun the VIF. Continue until there is no VIF value > 5.
#VIF values larger than 5 were found, so we must remove the largest and repeat.

vif(model.Prem_data.lm.2)

# Now we iteratively run the VIF function until no VIF values are greater than 5, removing the largest every iteration. Start with removing Attempted_Passes_per_game.
model.Prem_data.lm.2 <- lm(Prem_data$Points ~ Goals_per_game_inv + Assists_per_game_inv + Shots_per_game_sq + Shots_on_Target_per_game_sq + Completed_Passes_per_game_sq + Passing_Completion_Percent + Goals_Against_per_game_inv)

vif(model.Prem_data.lm.2)

# VIF values larger than 5 were found, so we must remove the largest and repeat.
```

```r
# Next remove Goals_per_game_inv
model.Prem_data.lm.2 <- lm(Prem_data$Points ~ Assists_per_game_inv + Shots_per_game_sq + Shots_on_Target_per_game_sq + Completed_Passes_per_game_sq + Passing_Completion_Percent + Goals_Against_per_game_inv)

vif(model.Prem_data.lm.2)

# VIF values larger than 5 were found, so we must remove the largest and repeat.
```

```r
Assists_per_game_inv 1.888053 Shots_per_game_sq 10.634158 Shots_on_Target_per_game_sq 7.996541 Completed_Passes_per_game_sq 14.727232 Passing_Completion_Percent 7.001382 Goals_Against_per_game_inv 4.019454
```
Now that all VIF values greater than 5 had been removed, we rechecked for a high Adjusted R-squared (above 0.85) and low p-value. Both of those conditions were met, so this linear regression model could be used as the model used to find our desired output.
Although the second model provided an output that could be used, we checked to see if a better model could be generated.

**Transformation & Third Model Creation**

A different sub-set of transformations were used for the creation of the third model. For the third model we had only modified 3 of the input parameters. We transformed the dataset in the following way:

- 3 variables have been transformed via a reciprocal model

Like the second model, the third model generated also had a high adjusted R-squared value (above 0.85) & low p-value, so we continued with this model to see if all 5 assumptions were satisfied.
Checking 5 Assumptions & Residuals – Model 3

After transformation, the third model needed to be checked to see if all assumptions were acceptable. In this case, all 5 assumptions are acceptable, so we could continue with this linear regression model. Because assumptions were met, we checked residuals plot as well to make sure it is a good fit of the dataset.
Checking & Removing Collinearity – Model 3

As we did for model 2, we again checked for collinearity between variables using VIF function for model 3.

The first run of VIF function for model 3 showed VIF values greater than 5, so we needed to do iterative process to remove largest VIF value per iteration. The images of the code below detail the process of how the largest VIF value was removed during each iteration until there were no VIF values greater than 5.
# VIF Check - Model 3

```
# If VIF > 5 for at least one variable, eliminate the variable with the highest VIF, and then iteratively rerun the VIF. Continue until there is no VIF > 5.

vif(model.Prem_data.lm.3)

# VIF values larger than 5 were found, so we must remove the largest and repeat.
```

<table>
<thead>
<tr>
<th>Goals_per_game_inv</th>
<th>Assists_per_game_inv</th>
<th>Shots_per_game</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.398774</td>
<td>39.902093</td>
<td>11.098295</td>
</tr>
<tr>
<td>Shoots_on_Target_per_game</td>
<td>Completed_Passes_per_game</td>
<td>Attempted_Passes_per_game</td>
</tr>
<tr>
<td>14.076427</td>
<td>1078.022152</td>
<td>968.635915</td>
</tr>
<tr>
<td>Passing_Completion_Percent</td>
<td>Goals_Against_per_game_inv</td>
<td></td>
</tr>
<tr>
<td>18.339095</td>
<td>4.154938</td>
<td></td>
</tr>
</tbody>
</table>

```
# Now we iteratively run the VIF function until no VIF values are greater than 5, removing the largest every iteration. Start with removing Completed_Passes_per_game.

model.Prem_data.lm.3 <- lm(Prem_data$Points ~ Goals_per_game_inv + Assists_per_game_inv + Shots_per_game + Shots_on_Target_per_game + Attempted_Passes_per_game + Passing_Completion_Percent + Goals_Against_per_game_inv)

vif(model.Prem_data.lm.3)

# VIF values larger than 5 were found, so we must remove the largest and repeat.
```

<table>
<thead>
<tr>
<th>Goals_per_game_inv</th>
<th>Assists_per_game_inv</th>
<th>Shots_per_game</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.248936</td>
<td>39.010701</td>
<td>9.545496</td>
</tr>
<tr>
<td>Shoots_on_Target_per_game</td>
<td>Attempted_Passes_per_game</td>
<td>Passing_Completion_Percent</td>
</tr>
<tr>
<td>13.507381</td>
<td>23.695066</td>
<td>15.535629</td>
</tr>
<tr>
<td>Goals_Against_per_game_inv</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.026414</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
# Next remove Goals_per_game_inv.

model.Prem_data.lm.3 <- lm(Prem_data$Points ~ Assists_per_game_inv + Shots_per_game + Shots_on_Target_per_game + Attempted_Passes_per_game + Passing_Completion_Percent + Goals_Against_per_game_inv)

vif(model.Prem_data.lm.3)

# VIF values larger than 5 were found, so we must remove the largest and repeat.
```

<table>
<thead>
<tr>
<th>Assists_per_game_inv</th>
<th>Shots_per_game</th>
<th>Shots_on_Target_per_game</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.164668</td>
<td>9.207490</td>
<td>8.885323</td>
</tr>
<tr>
<td>Attempted_Passes_per_game</td>
<td>Passing_Completion_Percent</td>
<td>Goals_Against_per_game_inv</td>
</tr>
<tr>
<td>23.537083</td>
<td>15.132432</td>
<td>4.820460</td>
</tr>
</tbody>
</table>
Now that all VIF values greater than 5 have been removed, we rechecked for a high Adjusted R-squared (above 0.85) and low p-value. Both of those conditions were met, so this linear regression model could be used as the model used to find our desired output.
Now that there are two working models (Model 2 & 3), we needed to decide which model to use. Because Model 3 had a higher Adjusted R-squared value, we choose this model for the linear regression.

Now that the model has been selected, it had been identified that the 4 remaining variables will be used as the inputs for the linear regression model:

- Assists per game (Inverse)
- Shots per game
• Passing Completion %
• Goals Against per game (Inverse)

Coefficients for the variables were then pulled and stored within another variable to be used in the calculation with the test dataset.

\[
\text{Coefficients for the variables were then pulled and stored within another variable to be used in the calculation with the test dataset.}
\]

**Test Model Prediction:**

The 2017-2018 Premier League season dataset is being used as the test dataset to check the model against. Shown as the output in the console window below are the 20 predicted point values for each of the teams from that season.

\[
\text{Test Model Prediction}
\]

\[
\text{The 2017-2018 Premier League season dataset is being used as the test dataset to check the model against. Shown as the output in the console window below are the 20 predicted point values for each of the teams from that season.}
\]
**Test Model Prediction - % Accuracy**

In terms of points, the model was able to predict the points accumulated by teams within the 2017-2018 season within the following accuracy bands:

- 7 team’s predictions within +/- 2 points
- 11 team’s predictions within +/- 4 points
- 17 team’s predictions within +/- 6 points

In terms of accuracy %, the model was able to predict the points accumulated relative to the number of total points that the team accumulated during the 2017-2018 season within the following accuracy bands:

- 6 team’s predictions within +/- 2.5%
- 12 team’s predictions within +/- 10%
- 17 team’s predictions within +/- 15%
We are able to extract additional information from the model when we plot the data against the predicted point difference as well as the % point difference at the same time by using color as an additional axis as shown below:

```
{r}
# Plotting the data
ggplot(Prem_data, aes(x=Prem_data_test$Points, y=Points.diff, color = Points.accuracy.pct)) +
    geom_point(alpha=1, size=5) +
    theme_bw() +
    scale_color_gradient2(low = "blue2", mid = "grey75", high = "red2")
```

What we see is that the most variance that we see within our model (from both a raw value difference & % difference) comes from teams that tallied less than 50 points within the 2017-2018 season. This makes sense as the same difference in predicted points between a team that only accumulated 40 points would contribute to be a larger total of the overall percentage than a team that accumulated 80 points within a season.

**Conclusion:**
Overall, the model was fairly accurate in being able to predict the number of points a team would accumulate based upon the model generated from 2018-2019 season data. The model that has been created now could be used to plug any premier league season (or other football league) dataset into for regression analysis.

We learned that it is important to generate a linear regression model systematically and trying multiple options to find the best fit for your dataset.

There are 2 main limitations to the model we created:

- Only 1 season worth of data was used to create the model. A larger sample size may yield an increase overall accuracy of the model.
- Each team may have their own style of play that may be captured in some but not all of the inputs that were used for the final linear regression model after collinearity checking. Teams that may play a style of football outside (or differs from) the variables that the linear regression model chose to use would most likely be the outlier datapoints that were less accurate than the rest.

**References:**

Data for the model generated within project was collected from the Premier League 2018-2019 Season.

Data for the test population in this project was collected from the Premier League 2017-2018 Season.

All data used within the project was collected from FBREF @ https://fbref.com/en/.