Model for Predicting Brownlow Medal Winner using Linear Regression

Author: Lang (Ron) Chen
Dec 2021 - Jan 2021

Abstract

The Brownlow Medal is the AFL's highest individual honour award. After each game the adjudicating umpire would award 3 votes, 2 votes and 1 vote to the top three performing players in their opinion, and the highest polling player of the whole season would be declared winner.

As the performance of AFL players are generally considered well represented by game statistics, the idea of ***training a Predictor*** with the goal of ***predicting the winner of the Brownlow Medal based on the data of all games that season*** came into existence.

There were four motivations for this project:

1. As an AFL enthusiast, the author hoped to train a Predictor which could to the best degree predict the Brownlow Medallist of 2022 and beyond.
2. Put the predictor training skills and theoretical knowledge learnt at University to test on a real world scenario.
3. Investigate how different choices made during predictor training (including some new innovative variations to traditional models) would affect performance statistically and empirically, so to aid future study and projects.
4. The Australian sports-betting industry is lucrative and the statistical/data science aspects of it is exciting; it is hoped a successful outcome could open the door to this industry for the author

The project was successful, and the resulting predictors can be found in *M1.ipynb*, *M2.ipynb* and *M3.ipynb*, and their details on page 10 of this report.

This report contains two parts:

Part 1: The model

Part 2: Empirical experiment
Part 2.1: Non PCA
Part 2.2: PCA

PART 1: THE MODEL

Choice of Underlying Model

The nature of the project and data available called for a supervised predictor model, which could both rank player performances in a particular game and also control how many players are predicted to receive votes for each game (i.e. for each game the predictor could only predict one player to receive three votes, one player to receive two votes and one player to receive one vote). Thus, with the limited choices of K Nearest Neighbour, Linear Regression and Decision Trees in the author's Data Science toolbox, ***(Multiple) Linear Regression was selected as the only viable choice*** *to satisfy the ranking condition*.

To satisfy the requirement of only predicting one player for each of three, two and one votes, it was decided that ***predictions must be performed simultaneously for all playerinstance[1] of the same game, rather than perform predictions for each playerinstance independently*** *(in other words, predictions would be made with on a 'game by game' basis rather than a 'player by player' basis for each player who took field in the game)*. This would allow for the top three 'scoring' (from Linear Regression) playerinstances from each game to be allocated votes, hence controlling the number of votes received by playerinstances from each game to a total of 6 votes.
(This would preventing cases like i.e. 2 players played wonderful games and thus were both predicted to receive 3 votes because the linear regression model output value for both of their playerinstances was more than 3)

Finally the votes would be counted up onto a leaderboard, just as the AFL does. The choice of this design utilised domain knowledge of the Brownlow Medal awarding system to ensure the process follows as closely to what is done in reality as possible, minimising potential systematic inconsistencies.

See PoC1 *(the .ipynb of Proof of Concept 1)*

However, the use of just a linear model was not satisfactory. Three more innovative variations of the Linear Model was also designed and included as a choice of model training for the project:

1. Bootstrapping (B):

   As the ratio of test case labels (0 votes:1:2:3) was generally (43:1:1:1), this Linear Regression Model variation would **bootstrap the playerinstances in the training data who are labelled 1 vote, 2 votes and 3 votes so their individual counts are equal to that of playerinstances with 0 votes** (ratio 1:1:1:1) before continuing on to train the model as per normal.

   > *i.e. bootstrapped training data would have same number of 0 votes playerinstances as before bootstrapping, while the number of 1 vote playerinstances would be the same as that of number of 0 vote playerinstances - and so forth for 2 votes and 3 votes)*

   See PoC2

[1] Playerinstance: Because each player plays many games and their stats of each game are stored as different and independent instances of data, the word playerinstance was defined to separate player from playerinstance (their performance/statistics in that game)

2. Linear Regression in Two Steps (2):

   Two Linear Models (LR1 and LR2) would be produced.

   Step 1: Use data as normal, except replacing the labels of 3, 2, 1 vote(s) to Boolean value of True, and 0 votes to False. Train the Linear Regression Model (LR1).

   Step 2: take just the 3, 2, 1 vote(s) playerinstances (labels as origin) and train a second Linear Regression Model (LR2)

   When inputting a game of data for prediction, take the top 3 scoring playerinstances in LR1 and input into LR2, then use the rank of LR2 score to assign prediction of 3 votes, 2 votes and 1 vote.

   See PoC3

3. Linear Regression in Two Steps with Bootstrapping (2)(B):

   Same as (2), except this time for step 1 use bootstrapping for the True labelled playerinstances as a whole.

   **note difference between bootstrapping in (2)(B) and (B): in (2)(B) the playerinstances are first relabelled and then bootstrapped. Thus the total number of playerinstances for trainingdata for LR1 should be half that of (B)*

   See PoC4

Method

1. Found websites containing game data of previous years suitable for scraping (footywire.com and AFLTables.com), and built crawler/scraper to scrape raw data of games from 2015 to 2021.
   1_CrawlAndScrape.ipynb *2_CrossValidation.ipynb*

2. Manipulated raw data into different forms[2]:
   - Normalised (N)
   - Standardised (S)
   - Percentage (P)
   - Rank [which is then] Standardised (RS)[3]

   Based on:
   - Both Teams/Whole Game (BT)
   - Own Team (OT)

   These would become some of the choices of Model Training later.
   3_AddDerivedAttributes.ipynb *4_DataManipulation.ipynb*

[2] Each column (i.e. handballs) would be manipulated (i.e. normalised/standardised etc) based on the data of either Both Teams and Own Team.

[3] As many models (including multiple linear regression) require standardised/normalised data, the ranks of the players based on a certain statistic had to be subsequently standardised.

Prepare Train-Test split data according to the 4 methods detailed above (i.e. Regular LR; (B); (2); (2)(B)). *(Seed 42 in Python was used for the entirety of this project)*

Note apart from test train split using games as unit, all of the 2021 season's data was put aside for empirical testing.

****Must remember to split data before performing bootstrapping/relabelling for (B), (2) and (2)(B)*

6_TestTrainDataPreparation.ipynb

3. With the many choices that could be experimented for training models, scripts with many layer of loops was coded and ran, to train and collect testing results on all permutation of choices.

   (If a model didn't return have any values that surpassed Feature Selection Value, then it would be marked as null.)

   Choices included:

   - Which Data Manipulation Method to use
     - N
     - S
     - P
     - RS
   - Which correlation coefficient cutoff value to use for feature selection (FS_Val)

     Used correlation of each column with label to determine whether to use in predictor. Any column with correlation above cutoff value would be accepted (as opposed to limiting the number of accepted columns).

     Values used were:
     - 0.2
     - 0.25
     - 0.3
     - 0.35
   - Whether to include Winloss column

     This was because Winloss was only stored as 0, 0.5 and 1 for loss, draw and win respectively, and is technically an ordinal categorical statistic. Thus at planning stage was unsure whether inclusion would make a differene to the model.

     - Include winloss (In)
     - Exclude winloss (Out)
   - How to balance Both Teams Data and Own Team Data (BT/OT)
     - 1. BT (only use Both Teams)
     - 2. OT (only use Own Team)

    - 3. BT_OT (for the same stat, only accept one of BT or OT to use in predicting) i.e. if ‘Handball OT’ had greater correlation with ‘Handball BT’, then the latter would not be selected even if it surpassed cutoff value)
    - 4. BT+OT (in the example above both would be accepted)

- Dealing with ‘Dependency Triangles’ (FS_Rule)

  There were many instances of statistics types that were dependent to each other. (i.e. Handball + Kicks = Disposals). In this case disposal is called a “sum statistic”. As some models require high degree of independence between the training attributes (a.k.a. the dependent variables in a linear model), different ways of dealing with Triangular Dependencies were experimented:

    - 1. All columns with correlation coefficient that passed Feature Selection cutoff value would be accepted (1)

    - 2. Strictly Independent (2)

      (i.e. if Disposals had highest correlation coefficient out of the three, then Handball and Kicks would both not be accepted even if they passed cutoff value)

    - 3. Remove Sum Statistics (3)

      (i.e. from beginning ban use of sum statistics such as disposals)

    - 4. Remove ‘Disposals’ only, others same as 2. (4)

      This was because Disposals often had a high correlation but limited many subcategories such as effective disposals, kicks etc which also had high correlation with the label

  **In data preparation process (3_AddDerivedAttributes.ipynb) filled out many of the ‘third corner’ of the ‘dependency triangle’ (derived statistics) – i.e. added ineffective disposals column which was part of the disposals and effective disposals ‘triangle.’ The ‘disposals efficiency%’ column was also converted to ‘effective disposals’*

- And of course, the five folds of KFold split

  The number of 5 was chosen because it ensures a large enough sample for training and testing (especially considering the approximately 3/46 = 6.5% proportion that 1 vote, 2 votes and 3 votes labelled playerinstances have within all data)

Data collected included:

1. Statistical Data
    - Regression Coefficient

      For (2) and (2)(B), the respective correlation coefficients for LR1 and LR2 would be stored separately

    - True Positive value for predicting players with 3 votes (tp3)

For (2) and (2)(B), the tp value of LR1 predicting vote-pollers as vote-pollers is stored as tp0.5

2. Emperical Data
   - Predicted top 20 players of the leaderboard from the 2021 full season test data, and their total "votes predicted to receive"

5_SetupDataCentre.ipynb Script1_LR.ipynb Script2_LR(B).ipynb Script3_LR(2).ipynb

Script4_LR(2)(B).ipynb

The experimental results (csv data outputted from the scripts) were renamed: R.csv, (2).csv, (B).csv, (2)(B).csv

**This process took a very long time – up to 40 hours of computational time to run all four scripts*

Results

Data Manipulation

When evaluating statistics, any combination of choices which contained one or more Fold where the model which failed to train (because no column passed the correlation cut-off value) was rejected because this highlights its instability - one particular way of splitting the train-test data could result in columns having extremely low predictive power.

Otherwise, groupby was used to group all folds of the same model together, and the data was manipulated into five Evaluation Statistics (the first two of which are traditional model Evaluation Statistics, whilst the latter three are quantifications of the empirical observations from the 2021 season data)

1. Correlation Coefficient (r):

   Took mean of the correlation coefficient of the five folds

   For cases of (2) and (2)(B), the average of the correlation coefficients of LR1 and LR2 were instead used.

   (Thus in a way, this Evaluation Statistic cannot be used to compare between (2) and non-(2) models)

2. TP3:

   Took mean of the True Positive value for predicting 3 votes.

   For tiebreaking purposes, can use TP2 and TP1 as secondary and tertiary sorting keys

3. M1(3):

   Actually consists two statistics:

   1. Score_RightPlace
      If the first placed player appeared in the top 3, then add

$$3 * (3 - abs(predicted\_place - 1))/3$$

to Score_RightPlace (initial value of 0).

If the second placed player appeared in the top 3, then add

$$2 * (3 - abs(predicted\_place - 2))/3$$

to the total score; and so forth for the third placed player.

Repeat this for all 5 folds and sum the individual Score_RightPlace to get the final Score_RightPlace for this model

2. Min_AvgVoteDiff
   For each top 3 player that was predicted to finish top 3 in a fold, record

$$abs(predicted\_votes - observed\_votes)$$

and take the average of all these VoteDiffs.
(Note: if top 3 player was not predicted to finish in top 3 in a fold, then that player will have NO VoteDiff value for that fold – i.e. it will not participate in the average calculation in the end)

Finally, rank by Score_RightPlace from largest to smallest, and using the Min_AvgVoteDiff as a tiebreaker key (ranking from smallest to largest)

This method is good because it doesn't force an integration of the two scores – which would be difficult to do as they measure different things.
Another important thing to note is that the two scores are rather independent from each other.

4. M1(4):
   Same as M1(3) except take the same scores for the top 4 players

   (i.e. formulae become

$$4 * (4 - abs(predicted\_place - 1))/4$$

$$3 * (4 - abs(predicted\_place - 2))/4$$

etc…)

5. M2:
   Same as M1(3) except only measure winning player's statistics

   (Score_RightPlace just becomes a tally of how many folds predicted the observed winner)

Model Selection

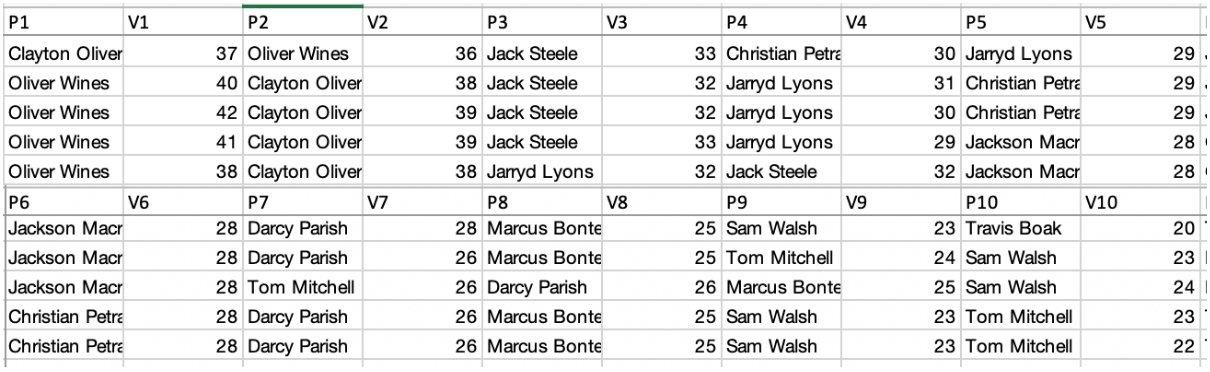
Three models were selected as Best and equal Second Best via the method of first listing the top 3 models based on each of the 5 Evaluation Statistics and then inspecting their predicted leaderboard:

Model number	Method	Data Manipulation Method	BT/OT	FS_Value	FS_Rule	Winloss inclusion
1	LR(B)	N	BT/BT_OT	0.2	2/4	(ALL)
2 (=2)	LR	N	BT/BT_OT	0.2	1	(ALL)
3 (=2)	LR	RS	BT/BT_OT	0.25	1	(ALL)

Of which:

- model 1 is the 3nd best model for correlation coefficient (and 3rd for M1(3));
- model 2 is the best model for M2;
- and model 3 is the best model for M1(3) (and 2nd for M1(4)).

Model 1 Leaderboard

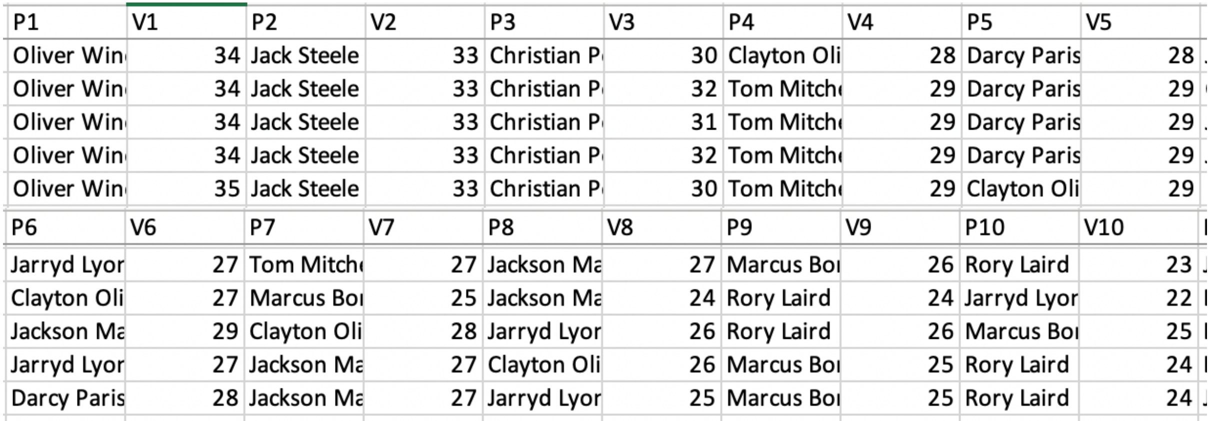


P1	V1	P2	V2	P3	V3	P4	V4	P5	V5
Clayton Oliver	37	Oliver Wines	36	Jack Steele	33	Christian Petra	30	Jarryd Lyons	29
Oliver Wines	40	Clayton Oliver	38	Jack Steele	32	Jarryd Lyons	31	Christian Petra	29
Oliver Wines	42	Clayton Oliver	39	Jack Steele	32	Jarryd Lyons	30	Christian Petra	29
Oliver Wines	41	Clayton Oliver	39	Jack Steele	33	Jarryd Lyons	29	Jackson Macr	28
Oliver Wines	38	Clayton Oliver	38	Jarryd Lyons	32	Jack Steele	32	Jackson Macr	28

P6	V6	P7	V7	P8	V8	P9	V9	P10	V10
Jackson Macr	28	Darcy Parish	28	Marcus Bonte	25	Sam Walsh	23	Travis Boak	20
Jackson Macr	28	Darcy Parish	26	Marcus Bonte	25	Tom Mitchell	24	Sam Walsh	23
Jackson Macr	28	Tom Mitchell	26	Darcy Parish	26	Marcus Bonte	25	Sam Walsh	24
Christian Petra	28	Darcy Parish	26	Marcus Bonte	25	Sam Walsh	23	Tom Mitchell	23
Christian Petra	28	Darcy Parish	26	Marcus Bonte	25	Sam Walsh	23	Tom Mitchell	22

Model 2 Leaderboard

Always predicts winner

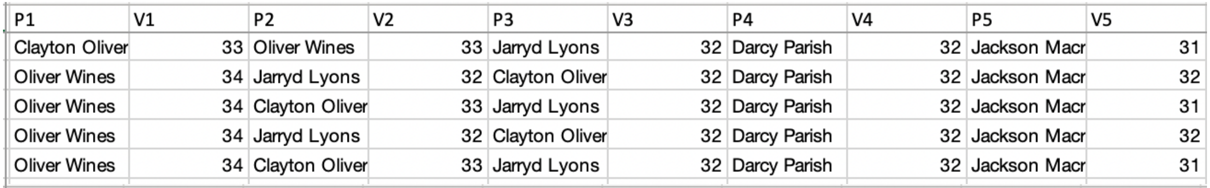


P1	V1	P2	V2	P3	V3	P4	V4	P5	V5
Oliver Win	34	Jack Steele	33	Christian P	30	Clayton Oli	28	Darcy Paris	28
Oliver Win	34	Jack Steele	33	Christian P	32	Tom Mitch	29	Darcy Paris	29
Oliver Win	34	Jack Steele	33	Christian P	31	Tom Mitch	29	Darcy Paris	29
Oliver Win	34	Jack Steele	33	Christian P	32	Tom Mitch	29	Darcy Paris	29
Oliver Win	35	Jack Steele	33	Christian P	30	Tom Mitch	29	Clayton Oli	29

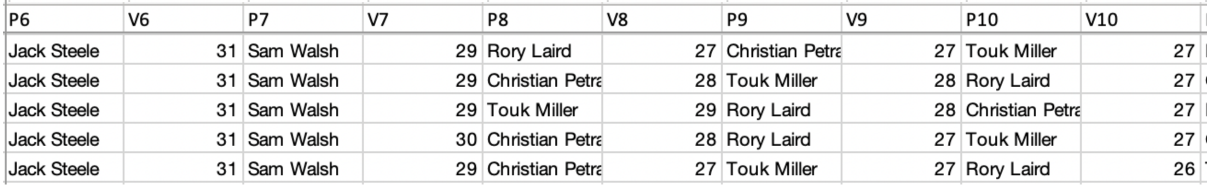
P6	V6	P7	V7	P8	V8	P9	V9	P10	V10
Jarryd Lyor	27	Tom Mitch	27	Jackson Ma	27	Marcus Boi	26	Rory Laird	23
Clayton Oli	27	Marcus Boi	25	Jackson Ma	24	Rory Laird	24	Jarryd Lyor	22
Jackson Ma	29	Clayton Oli	28	Jarryd Lyor	26	Rory Laird	26	Marcus Boi	25
Jarryd Lyor	27	Jackson Ma	27	Clayton Oli	26	Marcus Boi	25	Rory Laird	24
Darcy Paris	28	Jackson Ma	27	Jarryd Lyor	25	Marcus Boi	25	Rory Laird	24

Model 3 Leaderboard

Performs well for 4th and equal 5ths and has good performance for winner



P1	V1	P2	V2	P3	V3	P4	V4	P5	V5
Clayton Oliver	33	Oliver Wines	33	Jarryd Lyons	32	Darcy Parish	32	Jackson Macr	31
Oliver Wines	34	Jarryd Lyons	32	Clayton Oliver	32	Darcy Parish	32	Jackson Macr	32
Oliver Wines	34	Clayton Oliver	33	Jarryd Lyons	32	Darcy Parish	32	Jackson Macr	31
Oliver Wines	34	Jarryd Lyons	32	Clayton Oliver	32	Darcy Parish	32	Jackson Macr	32
Oliver Wines	34	Clayton Oliver	33	Jarryd Lyons	32	Darcy Parish	32	Jackson Macr	31



P6	V6	P7	V7	P8	V8	P9	V9	P10	V10
Jack Steele	31	Sam Walsh	29	Rory Laird	27	Christian Petra	27	Touk Miller	27
Jack Steele	31	Sam Walsh	29	Christian Petra	28	Touk Miller	28	Rory Laird	27
Jack Steele	31	Sam Walsh	29	Touk Miller	29	Rory Laird	28	Christian Petra	27
Jack Steele	31	Sam Walsh	30	Christian Petra	28	Rory Laird	27	Touk Miller	27
Jack Steele	31	Sam Walsh	29	Christian Petra	27	Touk Miller	27	Rory Laird	26

For comparison, the observed winners for 2021 were:

	1st	2nd	3rd	4th	5th	5th
Player	Oliver Wines	Marcus Bontempelli	Clayton Oliver	Sam Walsh	Darcy Parish	Jack Steele
Votes	36	33	31	30	26	26

However, one limitation is that there was no clear best model that can both

1. Predict the winner

2. Predict the top 4 very well;

It is believed this is either a limitation of the data or a limitation of the model used (or both)

Statistics for Final Models

Abiding strictly with the rules of Data Science, the final models were trained with a set of different data (i.e. not just picking the most favourable out of the KFold), even though the emperical observations (2021 season) on the final models did not look as impressive

	Model 1	Model 2	Model 3
r:	0.247	0.419	0.341
TP3:	0.489	0.233	0.137

```
[('Jack Steele', 39),
 ('Oliver Wines', 34),
 ('Clayton Oliver', 29),
 ('Marcus Bontempelli', 29),
 ('Christian Petracca', 28),
 ('Darcy Parish', 28),
 ('Jarryd Lyons', 26),
 ('Luke Parker', 24),
 ('Jackson Macrae', 21),
 ('Rory Laird', 21),
 ('Tom Mitchell', 20),
 ('Travis Boak', 20),
 ('Touk Miller', 19),
 ('Sam Walsh', 18),
 ('Jake Stringer', 18)]
```

```
[('Oliver Wines', 34),
 ('Jack Steele', 33),
 ('Christian Petracca', 30),
 ('Clayton Oliver', 28),
 ('Darcy Parish', 28),
 ('Jarryd Lyons', 27),
 ('Tom Mitchell', 27),
 ('Jackson Macrae', 27),
 ('Marcus Bontempelli', 26),
 ('Rory Laird', 23),
 ('Jake Stringer', 21),
 ('Cameron Guthrie', 20),
 ('Touk Miller', 20),
 ('Luke Parker', 20),
 ('Sam Walsh', 19)]
```

```
[('Clayton Oliver', 33),
 ('Jack Steele', 33),
 ('Oliver Wines', 32),
 ('Darcy Parish', 32),
 ('Jackson Macrae', 31),
 ('Jarryd Lyons', 30),
 ('Sam Walsh', 28),
 ('Christian Petracca', 27),
 ('Touk Miller', 27),
 ('Travis Boak', 23),
 ('Rory Laird', 23),
 ('Marcus Bontempelli', 23),
 ('Cameron Guthrie', 21),
 ('Dayne Zorko', 21),
 ('Zachary Merrett', 19)]
```

Qualitative/Empirical observations

1. Vote overshoot

   It was noticed that on many 'good' models the votes for the first 1-3 players went far into the 30s and even the 40s. One model (though likely ineffective) even reached 53 votes for a single player. From domain knowledge this is highly unreasonable as exceeding 30 votes marks an extraordinary season for a single player, and typically only the winner will exceed the 30 vote benchmark.

   However, it is notable that in recent years the winners have been polled higher than the historic average, with the 2015-2020 winners polling

   31, 35, 36, 28, 33, 31

   respectively. As we used the data from these years as training data, this perhaps can partly explain our overshoot. However, this is also an indication for the possible/likely presence of overfitting.

The actual empirical observations for votes looked as follows:

(choice of 36: historical maximum votes polled by any player in a single season)
(choice of 33: median value of winning poll from 2015-2020)

	<= 36	> 36	<= 33	>33
Maximum of votes polled by winners in all 5 models	244	464	42	646
Median of votes polled by winners in all 5 models	319	369	96	592

From the table above, it can be seen that more models overshoot than not, and thus all models trained using Linear Regression are systematically overshooting.

2. Systematic overpredicting/underpredicting for certain players

   In most of the leaderboards of the top models, Jack Steele (=5th), Christian Petracca, Jarryd Lyons and Jackson Macrae are players who systematically are ranked higher (and poll more votes) than the real 2021 leaderboard, whilst Marcus Bontempelli (2nd) and Sam Walsh (4th) are systematically ranked lower and receive less votes, with Bontempelli underpredicted by an average of approximately 5 votes and Walsh 10 – even within models which overshoot on voting.

   A suggested reason to explain this is that the selected columns used for prediction are the strengths of Steele and co, whilst not the strength of Bontempelli and Walsh.
   It is also arguable that as Macrae and Bontempelli are teammates, Macrae has 'stolen' a fair share of votes in prediction because his stats are closer to the 'average votepoller', but there are also other models which predict Petracca and Oliver (teammates) to both poll over 30 votes.

3. BT/OT

   Both Teams data has dominated the best models, with OT not appearing once in the top 3 models for all 5 Evaluation Statistics.

   **note BT/BT_OT practically equals BT (i.e. BT columns dominate their matching OT columns)*

4. Key statistics used to determine votes

   Model 1:

   Kicks, Handballs, Marks, Goals, Behinds, Tackles, Hitouts, Goal Assists, Inside 50s, Clearances, Clangers, Rebound 50s, Frees For, Frees Against (inverse), Contested Possessions, Uncontested Possessions, Effective Disposals, One Percenters, Bounces, Metres Gained, Turnovers, Intercepts, Time On Gorund %, Winloss, Behind Assists, Ineffective Disposals

Model 2:

Kicks, Handballs, Disposals, Goals, Inside 50s, Clearances, Contested Possessions, Uncontested Possessions, Effective Disposals, Centre Clearances, Stoppage Clearances, Score Involvements, Metres Gained, Behind Assists, Ineffective Disposals

Method 3:

Kicks, Disposals, Contested Possessions, Effective Disposals, Score Involvements, Metres Gained, Behind Assists

Suggested Future Improvements

1. After season 2022, can re-do this project and include 2021 data into the train_test mix

2. Could attempt to train model with data from more years (i.e. 2014 and before) – the only downside is that pre-2010 data on FootyWire.com does not have advanced statistics and 2010-2014 data does not have some of the advanced statistics (which all three models use).

PART 2: EXPERIMENTAL COMPONENT (EMPERICAL EXPERIMENT)

Introduction
'Empirical' in terms of the experiments on the effect of different choices refers purely to the fact that all results and conclusion drawn are from the data of different models trained for this specific project (and data), as opposed to using theoretical proof as the final resolution to the questions raised below. It should not be confused with the empirical Evaluation Statistics.

As a mathematical proof for truth cannot be completed by providing True examples, the experiments conducted here did not go into deep detail because regardless of how deep the analysis it still cannot be used to decisively prove any results. Nonetheless the observations here are still valuable and worth recording, just as empirical experiments are important in Computer Science.

> *An example of not going into detail: for feature cutoff value, could have recorded in detail how many 0.3 cutoff value > < = 0.2 for a specific Evaluation Statistic, and also for every pair of [0.2, 0.25, 0.3, 0.35] (6 permutations). However instead a simplified version of just collecting the cases where a larger cutoff value > < = a smaller cutoff value was recorded*

PART 2.1: NON-PCA RELATED

Data Manipulation Method

> ***Which Data Manipulation Method (N, S, P, RS) would be most 'useful' and thus should be the first choice for future projects?***

Observation 1:

> A subjective method of evaluation was used: look at the rankings of all models based on each of the 5 Evaluation Statistics and look at the order of appearance of each manipulation method starting from the top. (As the method starts from the top, even if 99% of models trained from data manipulated by this method is bad (due to inconsistencies with other choices) but 1% is excellent, this Data Manipulation Method would still be recognised accordingly)

Evaluation Statistic	1	2	3	4
r	N	P	S	>RS[4]
TP3	P	N	S	>>RS
M1(3)	RS	N	P	>S
M1(4)	N	RS	P	>S
M2	N	P	RS	>S

N performed the best out of the 4 methods, with P a close second and then RS, finally S is the worst performer.

[4] > means outperforms significantly; >> means outperforms extremely significantly

A suggested theoretical explanation for this is because Normalised data takes into account the sample variance, and thus captures the most of (or even does not lose any of) the information held by the sample.
On the other hand, Standardisation only takes into consideration of the min and max (and is thus sensitive to outliers), whilst Percentage only takes into consideration the sum (which is in some ways less susceptible to outliers and hence a better performance compared to S).
The performance of RS is a surprise (particularly given it made the top 3 final models) as technically Ranking 'throws away' the most information in the data. An explanation for this could be the widespread existence of outliers in the data, and thus RS (method most insensitive to outliers) was able to defeat S (the method most sensitive to outliers)

Conclusion: Overall this suggests that Normalisation should be the first choice of Data Manipulation Method for future projects, but the three other means (particularly P and then RS) should also be tried.

Conducting the same experiment on future projects will also likely lead to a more decisive conclusion on S and RS.

Observation 2:

Conclusion: It was discovered via empirical testing in excel (as part of the planning for this project) that the following Data Manipulation Methods are equivalent (as in final value would be the same if apply processes to same data):

- N = SN = PN
- S = NS = PS

**Note how equivalent Data Manipulation Methods will have its last operation the same as the 'single operation'.*

Unfinished experiments

- did not do RN (Rank followed by Normalisation); this could be worth a shot but the current hypothesis is that it would unlikely defeat any of R P or S.

Performance of (B), (2), (2)(B)

Did these models perform well enough to warrant future use and extension of theory onto other supervised models (i.e. KNN and DT)?

Observation 1:

Two sets of Data presented:

1. The top 3 models from ranking based on each of the 5 Evaluation Statistics:

Evaluation Statistic	**First**	**Second**	**Third**
r	B	B	B
TP3	B	B	B
M1(3)	LR	LR	B
M1(4)	LR	LR	B
M2	LR	LR	LR

Very clearly (B) dominated the theoretical statistics and made headways in the empirical statistics whilst LR comfortably led in the empirical statistics.

2. A subjective evaluation following the same logic as Data Manipulation Method's subjective evaluation:

Evaluation Statistic	**1**	**2**	**3**	**4**
r	B	>2B	>LR	>>2
TP3	B	LR	2B	2
M1(3)	LR	B	2	>2B
M1(4)	LR	B	2	> 2B
M2	LR	B	> 2	2B

(B) performed the best out of all, equal to R. Third place is (2)(B) and fourth place (2).

This is strong empirical evidence supporting the feasibility of the Bootstrapped method (seen by the convergence of two sets of data for the traditional statistics), which is hence suggestive that it could be extended for other supervised training methods.

The failure/unconvincing performance of (2) and particularly (2)(B) could be because of the lack of rows for training LR2. The training data for LR2 only consisted 2700 playerinstances comapred to LR1 (before bootstrapped) of more than 40000 playerinstances.

> Thus a further experiment of using more data from previous years (although they will contain less columns) should be conducted to see whether increasing the sample size will change the outcome, if a further empirical experiment for (2) and (2)(B) based on this project is deemed necessary.

Another reason for the unconvincing performance could be that the data just performs better for linear model training as a whole rather than training with just the vote-polling playerinstances. This could be partially explained by the fact that different AFL positions' 'good games' are characterised by different statistics (i.e. Forwards' goals, midfielder's disposals); sometimes influence on a game also simply does not get reflected fully in the statistics. This argument is supported by the fact that the award of votes is purely based on the head umpire's opinion immediately after the game, and he may be voting via his subjective judgement without ever referencing the stat charts.

A final and perhaps most decisive reason for the unconvincing performance of (2) is that quite frequently LR2 used only 1-2 columns, with one of them being Score Involvements and the other Behind Assists. Although domain knowledge deems the argument of 'Score Involvement being highly correlated to influence on a game and thus should be used to determine the allocation of the top 3 votes' reasonable, it seems unlikely that other domain-knowledge supported key statistics such as disposals play no part in the predictive process. Particularly in models that end up only using "Behind Assists" as the solidary attributes, this could be the primary reason for the bad results. Ultimately however, this problem could be traced back to the lack of instances for training LR(2) in the first place.

Conclusion: (B) can definitely be used (but its feasibility on empirical statistics should be continuously monitored), whilst (2) and (2)(B) may need further testing before a decisive conclusion can be made (currently deemed not successful). As for the question of applying the concept onto other supervised models, more testing needs to be conducted for all three variations.

**When using bootstrapped in future, ensure that the bootstrapped sample contains at least 1 of the original samples! i.e. save a copy of the pre-bootstrapped samples, and sample (required # of samples – len(pre-bootstrapped samples) to a new file and then concatenate the two. Although based on the chances of probability this flaw shouldn't have had a major impacts on the models trained or empirical experimental results in this project.*

Observation 2:

The actual output values of the final models themselves did not surpass 1.5 – nowhere close to 3.

The effect of the inclusion of winloss for a model

Hope to gain some idea of how a categorical variable would affect Linear Regression training, particularly for Data Manipulation Method like N and P which (typically) do not take range [0, 1]

Observation:

Using groupby to group models which are same except for inclusion of winloss, it was found that all 688 feasible model types trained demonstrated no difference between the inclusion or exclusion of winloss for all 5 Evaluation Statistics.

This likely means that winloss was not ever good enough to be used as a column in training.
This outcome deprived the oppotunity of investigating whether inclusion of a categorical variable would affect the model for good or for worse, particularly for models using normalised data (because the current data format of winloss is closer to being standardised data).

BT/OT rule

Investigating the effect of overfitting on Linear Regression Models.
Investigating whether strict assumptions of LR Models need to be upheld if output value is not what is looked for.

Observation:

The set of data is the top 3 models from ranking based on each of the 5 Evaluation Statistics

Evaluation Statistic	**First**	**Second**	**Third**
r	BT+OT	BT/BT_OT	BT/BT_OT
TP3	BT+OT	BT+OT	BT
M1(3)	BT/BT_OT	BT/BT_OT	BT/BT_OT
M1(4)	BT/BT_OT	BT/BT_OT	BT/BT_OT
M2	BT/BT_OT	BT/BT_OT	BT+OT

BT+OT leads in the traditional statistics, suggesting that dependencies 'enhance' the model in terms of traditional statistics.
However, the best models based on empirical statistics overwhelmingly use BT/BT_OT (meaning BT model = BT_OT model and hence BT_OT selects the same columns for training as BT).

In empirical selection, all of the TP3 models were not selected because of their poor empirical performance, whilst the Correlation Coefficient models better but had serious problem of overshooting.

Conclusion: This thus suggests that using dependencies are mostly not beneficial for empirical statistics and empirical observations (existence of degenerative overfitting).
However, the appearance of 'BT+OT' in third place of M2 is a sign that dependencies can be beneficial empirically, and this may be because the models only use the LR model output's rankings rather than output value.

This is very weak evidence to suggest that perhaps when not using LR for its output value or correlation coefficient, some restrictions on dependency can be loosened.

Dependency Triangle Feature Selection Rules

Investigating the effect of overfitting.

Investigating the way to resolve Dependency Triangles.

Observation:

Three sets of data presented:

Does rule 2's evaluation score rule method 1?

Evaluation Statistic	Yes	No	Equal	Total
r	0.000	0.955	0.045	176
TP3	0.739	0.216	0.045	176
M1(3)	0.330	0.466	0.205	176
M1(4)	0.375	0.477	0.148	176
M2	0.091	0.119	0.790	176

Does rule 3's evaluation score override rule 1?

Evaluation Statistic	Yes	No	Equal	Total
r	0.000	1.000	0.000	168
TP3	0.196	0.804	0.000	168
M1(3)	0.429	0.399	0.173	168
M1(4)	0.5	0.363	0.137	168
M2	0.155	0.143	0.702	168

Does rule 4's evaluation score override rule 1?

Evaluation Statistic	Yes	No	Equal	Total
r	0.000	1.000	0.000	168
TP3	0.738	0.262	0.000	168
M1(3)	0.363	0.452	0.185	168
M1(4)	0.411	0.464	0.125	168
M2	0.095	0.125	0.779	168

For the correlation coefficient, rule 1 either outperformed or drew with other rules.
For TP3, rule 2 and 4 outperformed rule 1 most times but this was the opposite for rule 3.
This shows that the models perform best on the most traditional statistic when allowed to have dependencies. But for the slightly more empirical leaning statistic of TP3, restricting dependency increases accuracy. (Rule 3's decrease for TP3 is likely because it bans Sum Statistics which have higher correlation with labels than their 'addend statistics')

However, peculiarly, for rule 2 and 4 all three empirical statistics suggests they prefer more the existence of dependency rather than not, thus quelling the argument that the allowance of all three statistics of a dependency triangle leads to excessive overfitting.
Rule 3 once again proved the opposite, preferring to not have dependencies, with no good reason to explain it.

Conclusion: TP3 does not benefit from the use of dependencies, whilst correlation coefficient either increases or is equal when using dependencies. For this project, empirical statistics seem to also prefer dependencies (thus no excessive overfit).

This is once again very weak evidence to suggest that perhaps when not using LR for its output value or correlation coefficient, some restrictions on dependency can be loosened.

Effect of increasing FS_Value

Investigating the effect of overfitting.

Investigating whether strict assumptions of LR Models need to be upheld if output value is not what is looked for.

Observation 1:

Does higher Feature Selection cutoff value lead to higher Evaluation Statistics?

Evaluation Statistic	**Yes**	**No**	**Equal**	**Total**
r	0.000	0.971	0.029	344
TP3	0.154	0.811	0.039	344
M1(3)	0.206	0.680	0.113	344
M1(4)	0.189	0.698	0.113	344
M2	0.052	0.291	0.657	344

Out of the instances when increasing the FS cut off value changes the model, all statistics prefer non-increase of FS_Value.
This suggests that the inclusion of more low correlation columns does not necessarily lead to overfitting

Conclusion: Increasing FS_Value diminishes both the traditional and empirical statistics (at least for this project). Future projects should use 0.2 or even 0.15 as a starting value.

Observation 2:

An error was made and amended during the experiment, whereby one of the scripts did not actually implement the FS_Value selection, meaning that all models trained utilised every single attribute column.
The results was that the error models came out extremely dominant for all 5 statistics but overshooting on votes was even more evident, further supporting the idea that decreasing FS_Value will improve models whilst not causing significant overfit.

Investigation on (2)

What value to denote 'True' for LR(1) training data?
Investigating whether strict assumptions of LR Models need to be upheld if output value is not what is looked for

Observation:

One aspect that was experimented on but not discussed in the methods section was the choice of the label value for True for LR with two stages. An experiment was conducted where all 'has vote' were re-labelled either 1 (minimum), 2 (mean), and 3 (max) (and False always labelled as 0), and different models based on these data were trained and compared.

Conclusion: Empirical observations demonstrated that keeping all other choices same, changing the value of True would produce the same model.

Theoretical backing for this observation is that because of the nature of Linear Regression models and the way it has been used for this experiment: instead of looking for a score, we looked for ranking only – and thus changing the slope of the best fit curve would not change the final rankings.

This seems to suggest that if the use of the Linear Model is only for ranking, rather than explicitly using the value of the output of the Linear Regression Model or its correlation coefficient, some of the strict assumptions can be relaxed.
Other inexplicit use of the Linear Model may also qualify for the relaxation of the strict assumptions.
**Note this observation was produced from a Multiple Linear Regression model!*

PART 2.2: PCA RELATED

Effect of increasing nComponents

Investigating whether increasing nComponents enhances the model.

Observation 1:

Evaluation Statistic	**nComponents**
r	10
TP3	10
M1(3)	9
M1(4)	9
M2	6/1

The top models for traditional statistics all use 10 components, suggesting that increasing nComponents will enhance the model based on traditional model Evaluation Statistics.
However, top models for empirical statistics also use high values of nComponents, and thus increasing nComponents benefits the model overall, rather than overfitting.

Conclusion: Increasing nComponents increases traditional Evaluation Statistics but doesn't seem to overfit

Observation 2:

Does increasing nComponents increase the Evaluation Statistics?

Evaluation Statistic	**Yes**	**No**	**Equal**	**Total**
r	1.000	0.000	0.000	1612
TP3	0.851	0.149	0.000	1612
M1(3)	0.473	0.418	0.109	1612
M1(4)	0.464	0.475	0.061	1612
M2	0.160	0.081	0.759	1612

Increasing nComponents 100% increases the Correlation Coefficient value, thus enhancing the model
Increasing nComponents mostly increases the TP3 value, thus also enhancing the value

For M1(3) and M1(4), the chances of enhancing the model by increasing nComponents is nearly the same as the chance of diminishing the model.
For M2, increasing nComponents mostly does not change affect the model, but when it does, the chance of enhancing the model is double the chance of dimishing the model.

This is a weak – if any - indication of overfitting (general correlation between performance on traditional statistics and empirical statistics closer to no correlation rather than negative correlation).

Conclusion: Increasing nComponents enhances the model based on traditional statistics, but for this project doesn't enhance the empirical statistics. Conclusions about overfitting cannot be drawn.

Effect of Increasing FS_Value

Investigating whether increasing Feature Selection cutoff Value enhances the model.

Observation:

Evaluation Statistic	0.3 > 0.2	0.2 > 0.3	Equal	Total
r	0.495	0.505	0.000	91
TP3	0.527	0.473	0.000	91
M1(3)	0.165	0.725	0.110	91
M1(4)	0.154	0.758	0.089	91
M2	0.011	0.286	0.703	91

Of the 91 models where both 0.2 and 0.3 produced models, the traditional statistics tend to show that changing the FS cutoff value doesn't significantly affect the model. However, the empirical statistics show that 0.2 tend to produce a better LR(PCA model)

Another warning is that there were 498 models in total, and the fact that in the end only 91 of them were from cutoff value of 0.3 suggests that 0.3 might often produce no model.

Conclusion: when using PCA, the cutoff value doesn't have to be as strict – PCA itself does contain feature selection properties so it might be best to let it do its job and not begin by throwing out columns which the PCA may be able to extract value out of. 0.2 might be a good starting point for future projects – even 0.15

*Note however that this experiment only compared 0.2 and 0.3, unlike section 2.1 which compared 0.2, 0.25, 0.3 and 0.35; so the results of the two experiments are not directly comparable

Ability for LR(PCA) to produce good models

Evaluating performance of PCA against non-PCA methods

Observation:

Evaluation Statistic	Best – Non-PCA	Best – PCA
r *	0.52086	0.2388
TP3	0.54988	0.44
M1(3)	17.67	17
M1(4)	27.5	28
M2	5	5

**Best came from (B), not (2), so it is comparable*

Statistically, none of the best LR(PCA) models could trump the non-PCA models based on traditional statistics, whilst for M1(4) it was able to slightly defeat the best Non-PCA model.

Empirically, looking at the leading pollers of the top models for the five Evaluation Statistics for both LR(PCA) and non-PCA, none of the best LR(PCA) models predicted as good as the top models for the non-PCA models.

Conclusion: PCA doesn't seem to perform well at least for this particular project (where the output value of the Linear Regression is not the directly used, and the Linear Regression model is only one part of the prediction process).

Shortcomings:
This was not exactly fair because (B), (2) and (2)(B) was not used for PCA, but the regular PCA models were compared against non-PCA models of (B), (2) and (2)(B).

LR(PCA) vs non-PCA LR

How well does PCA as a method compare against non-PCA (regular LR)

Observation:

Best LR(PCA) compared to LR

Evaluation Statistic	**PCA > LR**	**LR > PCA**	**Equal**	**Total**
r	0.295	0.007	0.679	352
TP3	0.273	0.048	0.679	352
M1(3)	0.156	0.122	0.722	352
M1(4)	0.153	0.134	0.713	352
M2	0.068	0.017	0.914	352

For both traditional and empirical Evaluation Statistics, the most frequent outcome is that PCA and LR produces the same model. This shows that PCA is indeed often capturing the same data as the using regular columns.

Out of the remaining cases, for traditional statistics PCA more often than not enhances the non-PCA models, whilst empirical statistics is closer to 50-50. Thus it shows that whilst PCA helps increase traditional statistics, at least in this project it is not overly helpful for improving the empirical statistics. Observing the leaderboard empirically, it is also evident that PCA models don't produce predictions that are as good as non-PCA models.

Conclusion: PCA is a helpful technique which can serve to enhance Linear Regression models, but on this project it didn't have a good effect.

*Note that before the process, PCA also required using Feature Selection values to do a pre-selection

BIBLIOGRAPHY

Data Source

AFLTables.com. 2022. *Brownlow Votes Round by Round*. [online] Available at: <https://afltables.com/afl/brownlow/brownlow_idx.html> [Accessed 26 January 2022].

Footywire.com. 2022. *AFL Fixture*. [online] Available at: <https://www.footywire.com/afl/footy/ft_match_list> [Accessed 26 January 2022].