Machine Learning Models

Data manipulation for machine learning

Data is collected and organised in two types, (i) game statistics, (ii) season results. Game statistics includes statistics of each team in a game, such as the offense, defense, field goal percentages, rebounds, assists, turnovers etc. Hence, each row in the data represents a single team in a match and its statistics. Season results data files include game-level results, i.e. each row corresponds to a game. The data file stores the day number, home and away teams and their scores. A description of the data file with combined game statistics and season results is displayed in Fig. 1.

Fig. 1. Game Statistics and Season Results

	Season	DayNum	TeamID	Score	OR	DR	Ast	TO	Stl	Blk	PF	Results	identifier	Loc	FGM3Rat	FGMRat	FTMRat
73730	2019	01	alabama	82	16	29	17	20	3	6	24	W	2019_01	H	0.38	0.49	0.68
73731	2019	01	arizona-state	102	20	38	11	17	3	2	26	W	2019_01	H	0.38	0.42	0.59
73732	2019	01	army-west-point	73	6	22	14	6	5	1	22	W	2019_01	H	0.30	0.43	0.69
73733	2019	01	auburn	101	19	23	24	13	9	8	21	W	2019_01	H	0.47	0.51	0.50
73734	2019	01	ball-state	86	16	26	14	9	7	5	18	W	2019_01	H	0.37	0.49	0.75
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
157148	2019	126	eastern-michigan	43	14	18	8	13	6	4	16	L	2019_126	H	0.24	0.29	0.50
157151	2019	127	jackson-state	49	9	21	10	11	11	6	18	L	2019_127	H	0.27	0.32	0.62
157152	2019	127	st-francis-pa	76	15	14	15	10	4	2	16	L	2019_127	H	0.52	0.48	0.24
157162	2019	129	unlv	55	11	32	12	10	2	4	23	L	2019_129	H	0.18	0.32	0.69
157164	2019	131	memphis	58	21	26	4	9	5	5	21	L	2019_131	H	0.17	0.24	0.85

4854 rows × 17 columns

Baseline Modeling: Logistic Regression Model Process

The outcome of home and away matches is a binary classification problem, i.e an observation must be classified as 0 (away win) or 1 (home win) . Based on the symmetry of this problem, a single model will be used to identify features that are important for a home team winning a game. This means that the same model can be used to identify features that are important for an away team to win a game, since if p is an estimation of the probability of a home team winning a game, then 1-p is an estimation of an away team winning a game.

The algorithm used to build the machine learning model is Logistic Regression. The algorithm requires the dependent variable (a.k.a the target) to have values denoting the classes being modeled. In this case, there are two classes as indicated above--0 denoting the away team winning and 1 denoting that the home team winning.

Since the models in this project are built under the assumption that each game has a home and away team, the games played in neutral sites will not be used in this project, and the dataset to be used in the model spans the seasons 2003- 2019.

Models built in this project use the ten in-game statistics shown in Fig 2. To model a given game, an aggregation of these statistics for prior games are computed, for both the home and away teams.The lagging period used for the prior game statistics to model a game is two games. This lagged aggregation method is used to avoid “leaking” information into the model that belongs to “the future”.

Fig 2: Features used for the analysis

Acronym	Description
Stl	Steals per team
Ast	Assists per team
TO	Turnovers per team
Blk	Blocks per match per team
PF	Personal fouls per team
OR	Offensive Rebounds per team
DR	Defensive Rebounds per match per team
FTMRAT	Field throw percentage
FGMRat	Field goal percentage
FGM3Rat	Three point percentage

To handle games at the beginning of the season, the design decision adopted in this project is to ignore the first two games.

An example of a prior game statistics being used as input to predict future games is illustrated in Fig. 3 and Fig. 4 below. In Fig. 3, the offensive rebound (OR) for "Duke" in row 562 (with value 17.5), is calculated by taking an average of the offensive rebound for rows 92 and 163 (with values of 19.0 and 16.0, respectively) in Fig. 4. The same method is applied for the rest of the features in Figures 3 and 4.

Fig 3: Data frame after two day moving average calculation

	Season	Score	OR	DR	Ast	TO	Stl	Blk	PF	FGM3Rat	FGMRat	FTMRat
92	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
163	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
562	2003.0	98.0	17.5	29.0	18.0	17.5	11.0	8.0	16.5	0.335	0.440	0.715

Fig 4: Original data frame before two day moving average calculation

	Season	DayNum	TeamID	Score	OR	DR	Ast	TO	Stl	Blk	PF	identifier	FGM3Rat	FGMRat	FTMRat
92	2003	019	duke	101	19	29	17	14	11	6	13	2003_019	0.35	0.43	0.67
163	2003	021	duke	95	16	29	19	21	11	10	20	2003_021	0.32	0.45	0.76
562	2003	033	duke	81	16	24	12	13	14	5	23	2003_033	0.17	0.48	0.62

Splitting of dataset for Training and Testing the Models

Dataset splitting allows checking for overfit issues, by splitting data into a training set, which is used to fit our model, and a test set, which is used to confirm that the model (fitted on training data) gives a similar rate of correct predictions on a new but similar data which was not part of fitting the model.

In this project, the games played by a team during 2003 - 2018 was used for training/fitting the model, while the data for games played during the 2019 season was held for testing.

The dataframe in Fig. 5, is split into two subsets, the 2003- 2018 seasons for feature selection and training, and the final season 2019 for testing.

It should be noted that due to the imbalance in the number of home and away games, the classification problem could have been resolved using a stratified train/test split. This splitting technique would have split the dataset into train and test sets, preserving the same proportions of examples in each class as observed in the original dataset.

The classification report and confusion matrix for the training and test dataset is displayed in Fig. 6 and 7.

Fig 5: Dataset with Home games

	Season	DayNum	TeamID	Score	OR	DR	Ast	TO	Stl	Blk	PF	Results	identifier	Loc	FGM3Rat	FGMRat	FTMRat
73730	2019	01	alabama	82	16	29	17	20	3	6	24	W	2019_01	H	0.38	0.49	0.68
73731	2019	01	arizona-state	102	20	38	11	17	3	2	26	W	2019_01	H	0.38	0.42	0.59
73732	2019	01	army-west-point	73	6	22	14	6	5	1	22	W	2019_01	H	0.30	0.43	0.69
73733	2019	01	auburn	101	19	23	24	13	9	8	21	W	2019_01	H	0.47	0.51	0.50
73734	2019	01	ball-state	86	16	26	14	9	7	5	18	W	2019_01	H	0.37	0.49	0.75
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
157148	2019	126	eastern-michigan	43	14	18	8	13	6	4	16	L	2019_126	H	0.24	0.29	0.50
157151	2019	127	jackson-state	49	9	21	10	11	11	6	18	L	2019_127	H	0.27	0.32	0.62
157152	2019	127	st-francis-pa	76	15	14	15	10	4	2	16	L	2019_127	H	0.52	0.48	0.24
157162	2019	129	unlv	55	11	32	12	10	2	4	23	L	2019_129	H	0.18	0.32	0.69
157164	2019	131	memphis	58	21	26	4	9	5	5	21	L	2019_131	H	0.17	0.24	0.85

4854 rows × 17 columns

Fig 6: Confusion Matrix and Classification Report for Training Set

```
(73389, 10)


[[ 2535 22432]
 [ 2181 46241]]


[Train Classification Report]
              precision    recall  f1-score   support

           0       0.54      0.10      0.17     24967
           1       0.67      0.95      0.79     48422

    accuracy                           0.66     73389
   macro avg       0.61      0.53      0.48     73389
weighted avg       0.63      0.66      0.58     73389


[Train] Accuracy score (y_predict_training, ytrain_home): 0.6646227636294267
```

Fig 7: Confusion Matrix and Classification Report for Test Set

```
(4854, 10)


[[ 225 1488]
 [  10 3131]]


[Test Classification Report]
              precision    recall  f1-score   support

           0       0.96      0.13      0.23      1713
           1       0.68      1.00      0.81      3141

    accuracy                           0.69      4854
   macro avg       0.82      0.56      0.52      4854
weighted avg       0.78      0.69      0.60      4854


[Test] Accuracy score (y_predict_test, ytest_home): 0.6913885455294603
```

Interpretation

The closeness of the training and the test accuracy score shows that the model will "generalize well", and that the model will predict well , when new data is presented to it.

Accuracy alone is not a reliable metric to assess the performance of the logistic regression model, a classification report was used to assess the performance of the model.

The classification report for Fig. 16 has 73389 predictions for the home games, and out of the predictions , the confusion matrix predicted 22432+46241 times for home wins, while predicting 2535+2181 wins for away games. While in reality there were 2535+22432 away wins, and 2181+46241 wins in home matches.

However, in the classification report for the test set in Fig. 17 the total the model made 4854 predictions for the home games, out of these predictions, the confusion matrix predicted wins **1488+3131** times for home games, while predicting **225+10** wins for away games. While, in reality there were **225+1488** away wins, and **10 + 3131** wins in home matches.

The precision value for the minority class (0) tells us how often our model is correct when predicting away wins, while the recall value for the majority class (1) tells us out of all the home wins, how many did our model correctly identify.

For the minority class of the classification report for both the training and test data, a high precision value and a low recall value was obtained. This is basically saying that the model is correctly predicting a high percentage of the time. However, of all the away wins in the dataset, the model did not not catch many of them, hence the low recall value.

Extended Modeling: Feature Selection by Recursive Feature Elimination (RFE)

The original model is extended by using a feature selection algorithm called Recursive Feature Elimination (RFE). The feature selection was done to reduce the number of features needed to predict the outcome of matches, while also maintaining the accuracy of the model. The feature selection process would also tell us which features contribute most to the outcome of a home team winning or losing a match. The RFE selection process works by using the model accuracy to identify which combination of attributes contribute most to predicting the outcome of matches i.e. home win or away win. The RFE algorithm works by fitting the machine learning algorithm used in the model **(logistic regression)**, ranking features used in the model by importance, and discarding the least important features, and re-fitting the model [1]. The description of the features used for the data modeling is in Fig. 2. From these features, the RFE algorithm was used to select the five most important features contributing to the outcome of matches for home games below.The result of the feature selection method can be seen in the classification report of Fig. 8 . These features were assigned a value of 1 and displayed in Fig. 9

[1] "Recursive Feature Elimination (RFE) for Feature Selection in ...." 25 May. 2020, https://machinelearningmastery.com/rfe-feature-selection-in-python/. Accessed 14 Jul. 2020.

Fig 8: Confusion Matrix and Classification Report for Test Set, after building model with the features selected by RFE

```
[Confusion Matrix for Test Data using RFE]
[[ 247 1466]
 [  15 3126]]


[Test Classification Report]
              precision    recall  f1-score   support

           0       0.94      0.14      0.25      1713
           1       0.68      1.00      0.81      3141

    accuracy                           0.69      4854
   macro avg       0.81      0.57      0.53      4854
weighted avg       0.77      0.69      0.61      4854


[Test] Accuracy score (y_predict_test, ytest_home): 0.6948908117016893
```

Comparing Full Model vs. Reduced Model

Comparing the performance metrics from Fig. 7 and 8, the accuracy score was slightly improved (0.694 vs. 0.691) when the top five features (reduced model) were used to predict the classifier accuracy.

The full model does a better job predicting away wins versus the reduced model. This is because the precision value for the minority class for the full model is 0.96 versus 0.94 for the reduced model. This shows us that the original 10 features are better predictors of away wins.

These top five features as denoted by the ranking by the ranking of 1 in the RFE algorithm in Fig. 9 are : FGM3Rat, FGMRat, OR, FTMRat and BLK.

Fig 9: Feature Rankings for Home Games

	Feature	Ranking
0	OR	1
5	Blk	1
7	FGM3Rat	1
8	FGMRat	1
9	FTMRat	1
3	TO	2
4	Stl	3
1	DR	4
2	Ast	5
6	PF	6

Conclusions

Logistic Regression was used to build models to predict the outcome of matches in NCAA basketball games for 351 different teams. More specifically, the models estimate the probability of a home team winning a match. Data was partitioned into training and test for model validation. The training set included data from 2003-2018, and the test data was for the 2019 season. A performance matrix was generated both for the training and test data.

The global accuracy prediction for the test and training data was sixty nine percent for model prediction. However, for the performance matrix for both the training and test data, a low recall value was obtained for the minority class (away wins) versus the recall value obtained for the majority class (home wins).

This implies that the model is good when predicting when the home team is going to win, and not good at predicting when an away team is going to win

From the perspective of improving the global accuracy of the Logistic Regression models using RFE, the most important features are: field goal percentage made, field throw percentage made, free throw percentage made, blocks and offensive rebounds. This **does not** mean that these features are the most important ones with respect to a home team winning a game.

In order to determine which features contribute the most to the home teams winning a game, through a Logistic Regression (LGR) model, would be to analyze the coefficients of the LGR models.