Sudarat Sukjaroen student Id 24667255

Assignment 1 - Kaggle Competition

The CRISP-DM reference Model

1. Business understanding

1.1 Determine business objectives

The NBA draft is an annual event in which teams select players from their American colleges as well as international professional leagues to join their rosters. Moving to the NBA league is a big deal for any basketball player. Data science is tasked to build a model that will predict if a college basketball player will be drafted to join the NBA league based on his statistics for the current season.

1.2 Determine data mining goals

The primary goal is to build a predictive model that determines whether a college basketball player will be drafted into the NBA based on their current-season statistics. This involves selecting relevant features, evaluating model performance, ensuring interpretability, and considering ethical implications. Once developed, the model can be integrated into NBA scouting processes and continuously monitored and updated for accuracy and fairness.

1.3 Project scope

There are 4 weeks to run the experiment and summarise the result every Friday. Each experiment includes data, preparation, Feature Engineering, Modelling, and Technical performance.

2. Data understanding

2.1 Collect initial data

Reading Data from CSV Files

- Number of training data set = 56,091 (total 64 columns)
- Number of test data set = 4,970 (total 63 columns)

2.2 Describe data

Table 1 : Training Data description

No.	Feature name	Data type	Null count	Zero Count
1	num	object	4,669	415
2	yr	object	274	0
3	ht	object	80	0
4	team	object	0	0
5	conf	object	0	0
6	type	object	0	0
7	player_id	object	0	0
8	TPM	int64	0	18,222

No.	Feature name	Data type	Null count	Zero Count
9	TPA	int64	0	11,709
10	FTM	int64	0	7,474
11	FTA	int64	0	6,272
12	twoPM	int64	0	6,082
13	twoPA	int64	0	3,254
14	GP	int64	0	0
15	year	int64	0	0
16	pick	float64	54,705	0
17	Rec_Rank	float64	39,055	0
18	dunks_ratio	float64	30,793	1,077
19	mid_ratio	float64	9,688	4,664
20	rim_ratio	float64	9,464	2,119
21	dunksmade	float64	6,081	25,789
22	dunksmiss_dunksmade	float64	6,081	24,712
23	midmade	float64	6,081	8,271
24	rimmade	float64	6,081	5,502
25	midmade_midmiss	float64	6,081	3,607
26	rimmade_rimmiss	float64	6,081	3,383
27	ast_tov	float64	4,190	3,258
28	drtg	float64	44	0
29	adrtg	float64	44	0
30	dporpag	float64	44	0
31	stops	float64	44	0
32	bpm	float64	44	0
33	obpm	float64	44	0
34	dbpm	float64	44	0
35	gbpm	float64	44	0
36	ogbpm	float64	44	0
37	dgbpm	float64	44	0
38	blk	float64	38	14,333
39	stl	float64	38	7,491
40	ast	float64	38	6,286
41	oreb	float64	38	6,045
42	dreb	float64	38	3,272
43	pts	float64	38	3,175
44	treb	float64	38	2,508
45	mp	float64	38	19
46	TP_per	float64	0	18,222
47	blk_per	float64	0	14,823
48	stl_per	float64	0	8,000
49	FT_per	float64	0	7,474
50	AST_per	float64	0	6,831
51	ftr	float64	0	6,553

No.	Feature name	Data type	Null count	Zero Count
52	ORB_per	float64	0	6,495
53	twoP_per	float64	0	6,082
54	eFG	float64	0	4,605
55	TO_per	float64	0	4,576
56	DRB_per	float64	0	3,670
57	TS_per	float64	0	3,661
58	pfr	float64	0	3,479
59	Ortg	float64	0	2,490
60	usg	float64	0	1,158
61	adjoe	float64	0	68
62	Min_per	float64	0	23
63	porpag	float64	0	0
64	drafted	float64	0	55,555

Note: Metadata has 70 features, but after re-recheck, the feature names are the same as the training and testing data set, but the numbers 24-26, 51-53, and 65-70 are missing, probably because of data privacy concerns.

There are 23,740 unique player IDs in the training data set and 4,968 in the testing data set. Then, map the player ID as an integer number name player numbers to create a scatter graph to see the data distribution.

2.3 Verify data quality

1) Check and drop duplicated data
   Not found duplicate data in the training and testing data set.
2) Check the Null value in the column
   As seen in the data description table

2.4 Generate test design

The experiment will consider these metrics values:

1) Accuracy = (TP + TN) / (TP + TN + FP + FN)
2) Precision = TP / (TP + FP)
3) Recall = TP / (TP + FN)
4) F1 score = 2 * (precision * recall) / (precision + recall)

The metrics 1-4 have perfect scores 1.

Experiment week 1

1. Data preparation

1) Clean data
   replace Null value to 0
2) Format data Mapping column from text to number
   a) Team to team number
   b) Conf to conf_number
   c) Yr to yr_number
   d) Ht to ht_number
   e) Num to num_number
   f) Player_id to player_number

2. Feature engineering

1) Calculate Information Gain (IG) - feature selection processes to determine the importance of features in a dataset. But in the first experiment, I decided to use all the features.

Table 2 : Calculate Information Gain (IG)

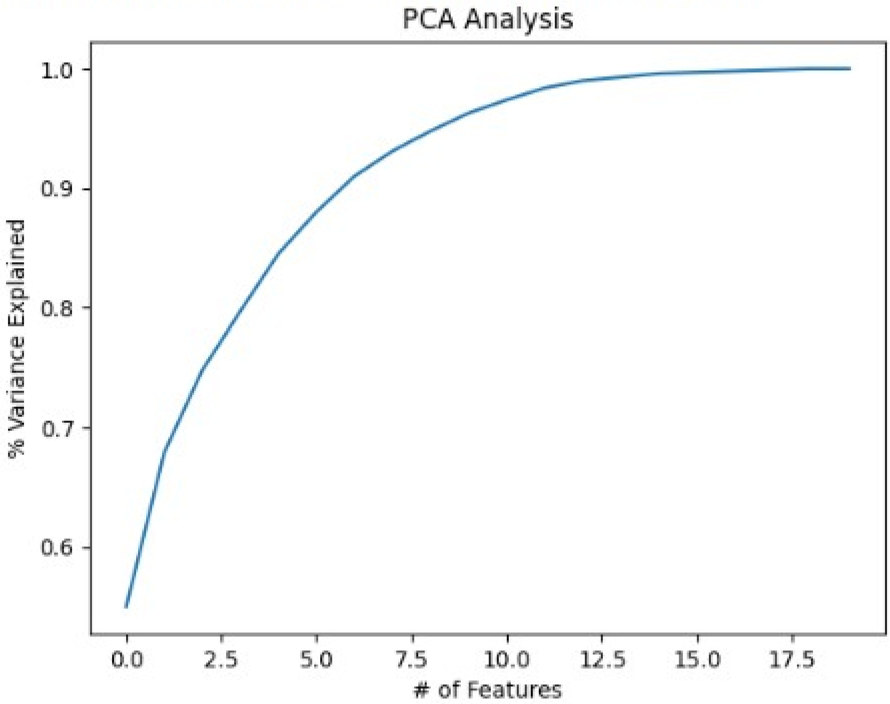
No.	Feature name	IG value	Description
1	pick	0.03769	Order of NBA draft
2	dporpag	0.02318	Asdjusted porpag
3	porpag	0.02212	Points Over Replacement Per Adjusted Game
4	gbpm	0.02141	BPM 2.0
5	bpm	0.01928	BPM - Estimate the player’s contribution in points above league average per 100 possessions played
6	stops	0.01891	Stops - Stops; Dean Oliver's measure of individual defensive stops
7	adjoe	0.01847	AdjO – Adjusted offensive efficiency
8	ogbpm	0.01846	Offensive BPM 2.0
9	twoPM	0.01664	2P - 2-Point Field Goals
10	pts	0.01648	PTS - Points
11	Rec_Rank	0.01635	Recruiting rank
12	twoPA	0.01562	2PA - 2-Point Field Goal Attempts
13	obpm	0.01524	Offensive BPM
14	FTM	0.01449	Free Throws
15	FTA	0.01431	Free Throw Attempts
16	dreb	0.01313	DRB - Defensive Rebounds
17	team_number	0.01224	Name of team
18	mp	0.01203	MP - Minutes Played
19	treb	0.01154	TRB - Total Rebounds
20	adrtg	0.01148	Adjusted DRtg
21	rimmade	0.01137	Shots made at or near the rim
22	GP	0.01134	Games played
23	midmade_midmiss	0.01129	Sum of Two point shots that were not made at or near the rim and Shots missed
24	dunksmade	0.01066	Dunks made
25	midmade	0.01054	Two point shots that were not made at or near the rim
26	rimmade_rimmiss	0.01021	Sum of Shots made at or near the rim and Shots missed

Note: the List of features with an IG value of more than 0.01 and their description.

2) Calculate Principal Component Analysis (PCA) dimensionality reduction technique
Calculate PCA 80%, 90%, and 100%, but the number of components for 80% and 90% is almost the same as 100%. Then, I decided to use 100% to experiment.

Figure 1 : Result from Principal Component Analysis (PCA))

```
Number of classes =  2
[5.50240199e-01 1.28644919e-01 6.88809067e-02 4.92114822e-02
 4.77420077e-02 3.52240262e-02 3.01384645e-02 2.10874566e-02
 1.74694045e-02 1.49650228e-02 1.09334084e-02 9.53722859e-03
 5.61017741e-03 3.19451562e-03 2.73398811e-03 1.47590619e-03
 1.21503028e-03 7.42851126e-04 5.60214537e-04 3.92791138e-04]
```




```
[0.55  0.679 0.748 0.797 0.845 0.88  0.91  0.931 0.948 0.963 0.974 0.984
 0.99  0.993 0.996 0.997 0.998 0.999 1.    1.   ]
Number of Components (>Z%) =  6
Data Shape =  (56091, 70)
```

3. Modeling

Logistic regression - Find the best hyperparameter set
Splite data percentage 80:20
Best Hyperparameters ‘C’ = 1, 'penalty' = ‘l2’

4. **Evaluation**

Figure 2 : Confusion Matrix week 1

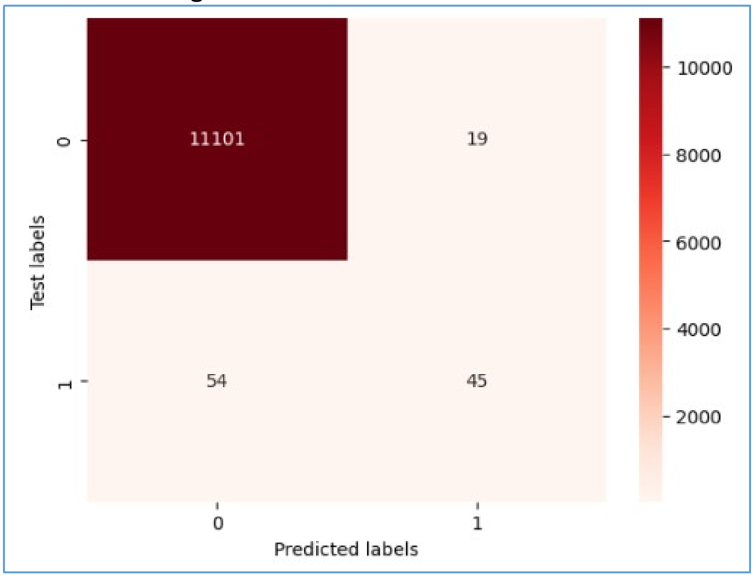
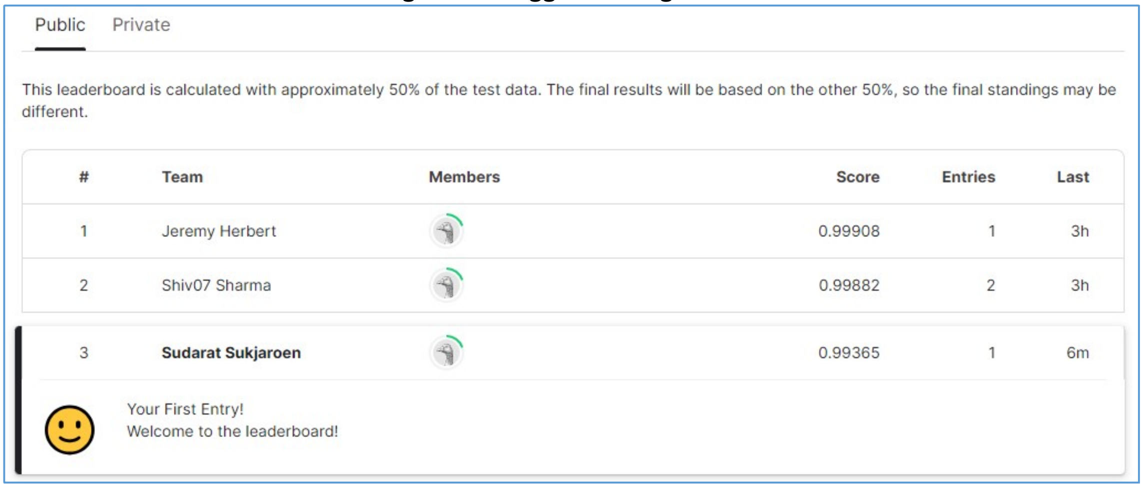


Table 3 : Metric values week 1

Week No.	Accuracy	Precision	Recall	F1 Score
1	0.99349	0.70313	0.45455	0.55215

Figure 3 : Keggle ranking week 1



<u>Plan for experiment in Week 2 - 4</u>

In the following experiments, I plan to apply more machine-learning techniques.

1. Run other classification models.
2. Replace the Null value with Means, Median, Mode, etc.
3. Remove outlier values.
4. Replace missing value with Means, Median, Mode, etc.
5. Adjust the density of the ratio of drafted = 0 and 1.
6. Apply Information Gain (IG).
7. Apply Principal Component Analysis (PCA).

All techniques from Accuracy, Precision, Recall and F1 Score values will be considered before run prediction.

Experiment week 2

Basic data preparation, Feature engineering, and Modelling are the same as in Week 1. The accuracy benchmark is more than 0.99349 to include in the final experiment.

Feature engineering

SMOTE (Synthetic Minority Over-sampling Technique) is an oversampling method used in machine learning to balance class distribution by generating synthetic samples of the minority class.

Also, create player_number to be a unique id from player_id to create a scatter graph to represent data distribution.

Table 4 : Accuracy value from Logistic regression model and Smote

No of parameters	Smote	Accuracy
C = 0.1 class_weight = None max_iter = 300 penalty = l2	Yes	0.98928
C =10 penalty = l2	Yes	0.98553
C = 1 penalty = l2	Yes	0.98503

Data preparation

Clean data object type feature name 'yr' and 'ht' and filter outlier feature order by Information Gain. Yellow highlight is included in the final experiment.

Table 5 : Accuracy value after filter ouliter from features

No.	Outlier	Accuracy
1	yr	0.99391
2	ht	0.99210
3	pick	0.78417
4	dporpag	0.99365
5	porpag	0.99347
6	gbpm	0.99221
7	bpm	0.99266
8	stops	0.99311
9	ogbpm	0.99338
10	adjoe	0.99291
11	pts	0.99168
12	Rec_Rank	0.98210
13	twoPM	0.98930
14	twoPA	0.99349
15	obpm	0.99305
16	FTM	0.99251
17	FTA	0.99167
18	dreb	0.99162
19	mp	0.99349

Figure 4 : Example of scatter graph before filter outlier from bpm

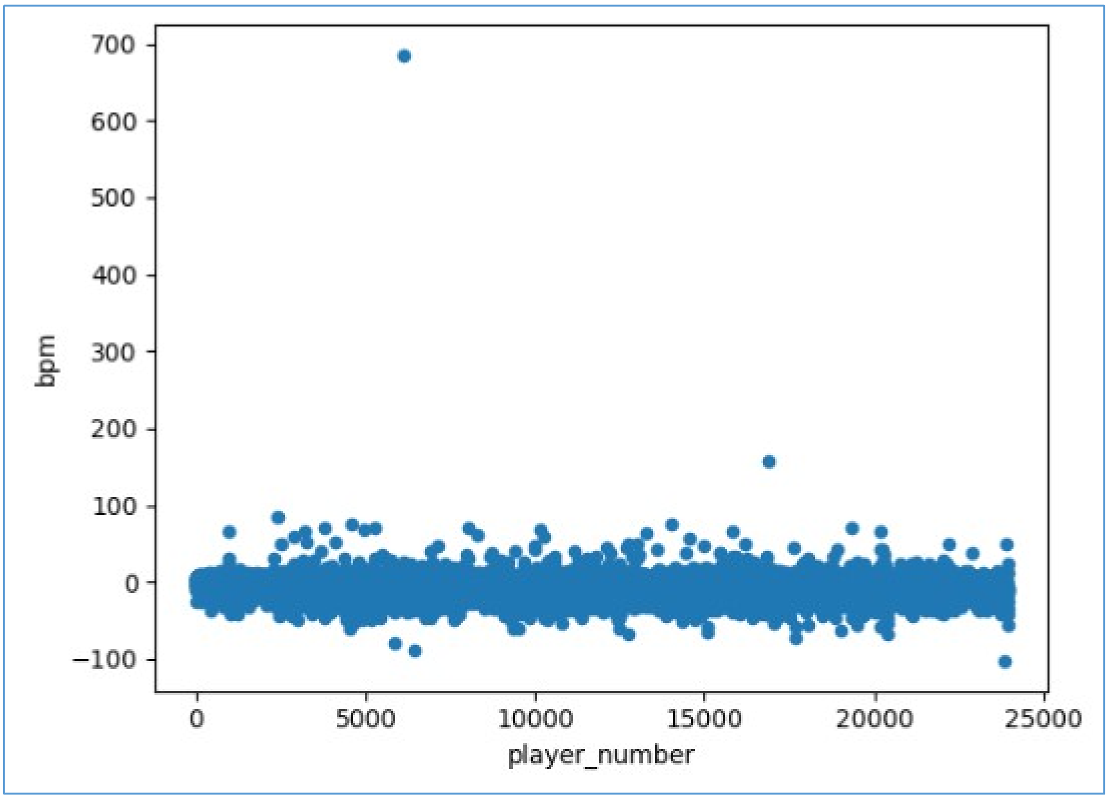


Figure 5 : Example of scatter graph after filter outlier from bpm

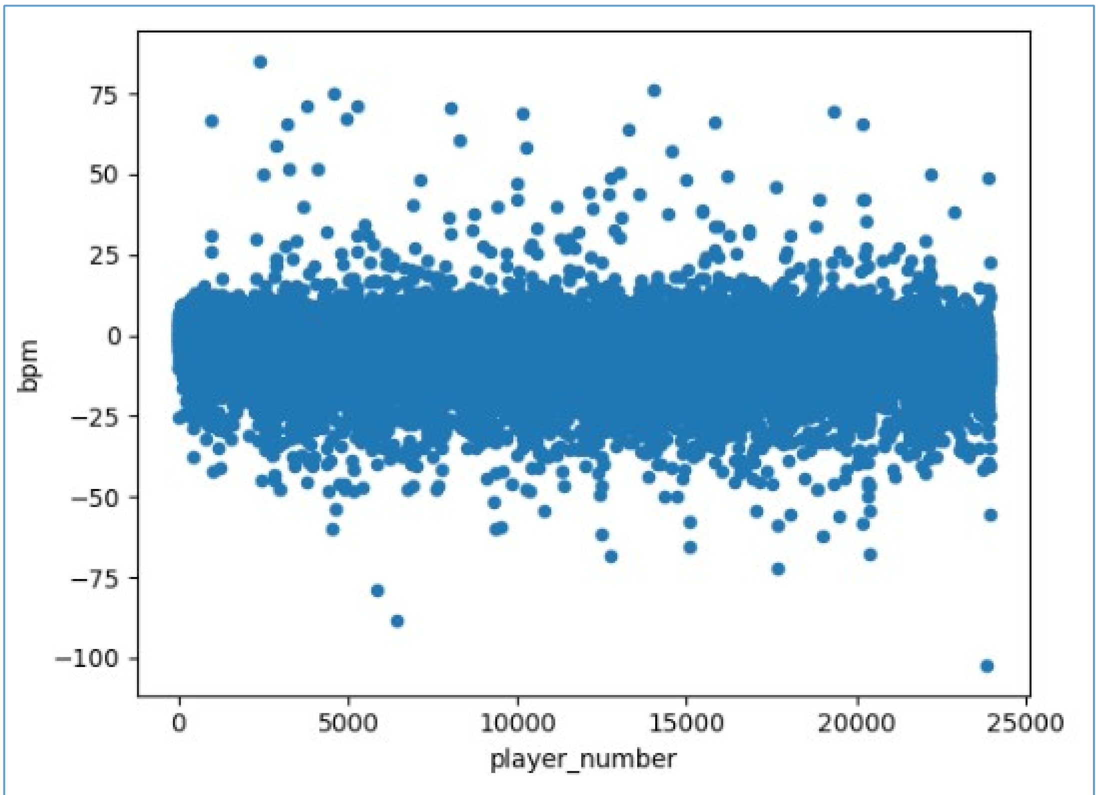


Figure 6 : Example of scatter graph before filter outlier from twoPA

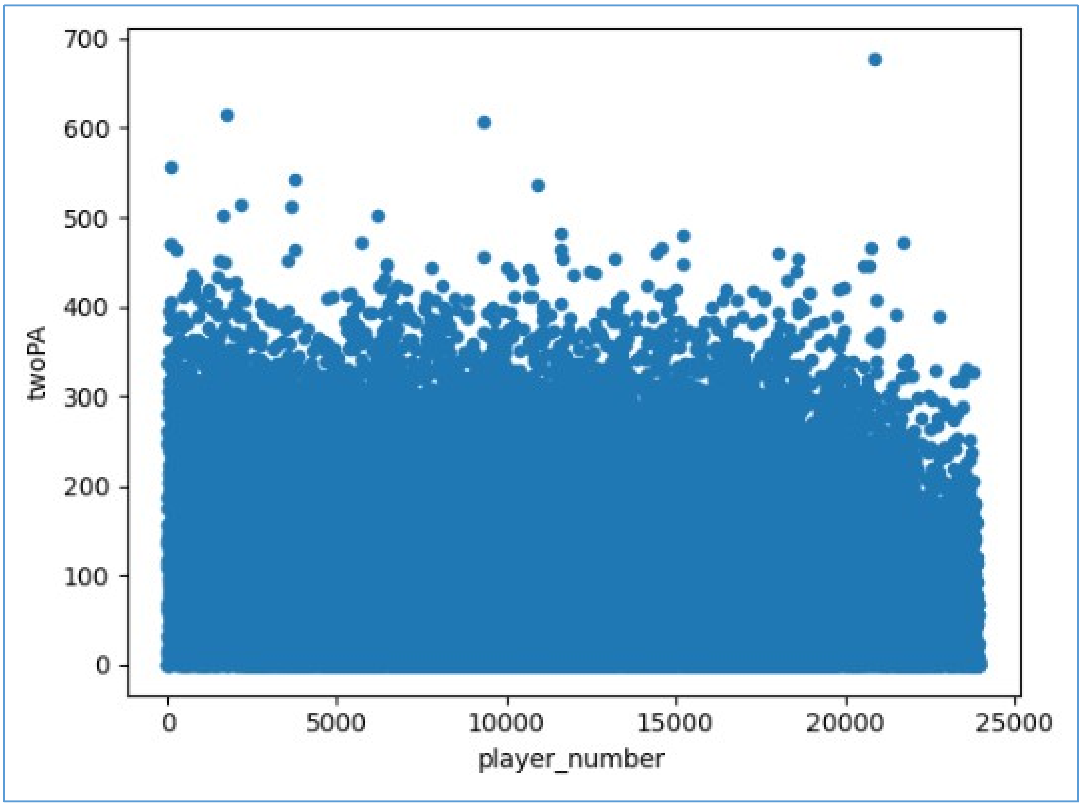


Figure 7 : Example of scatter graph after filter outlier from twoPA

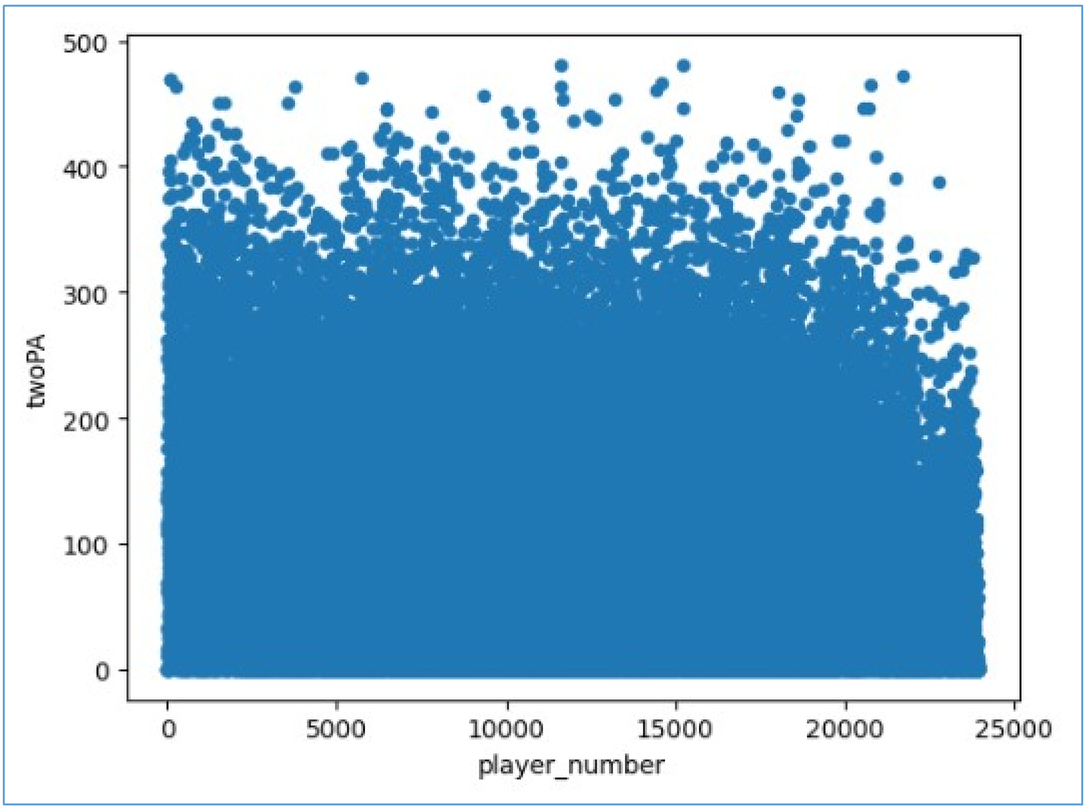


Figure 8 : Confusion Matrix week 2

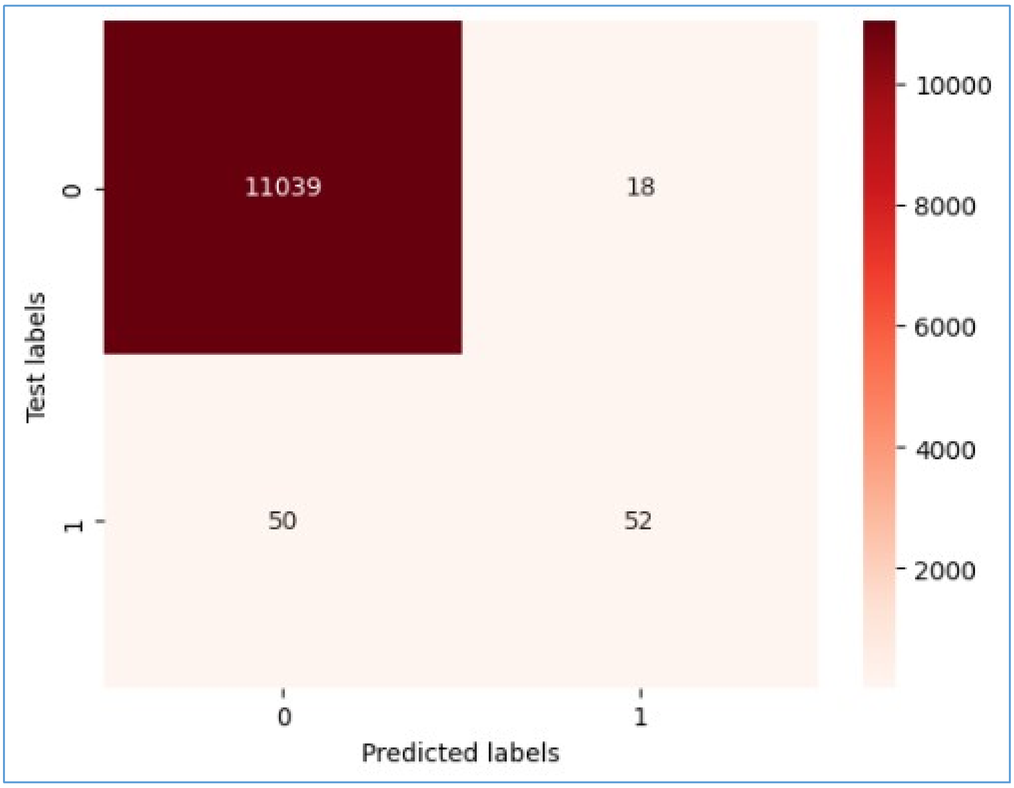


Table 6 : Metric values week 2

Week No.	Accuracy	Precision	Recall	F1 Score
1	0.99349	0.70313	0.45455	0.55215
2	0.99391	0.74286	0.50980	0.60465

Experiment Week 3

Basic data preparation, Feature engineering, and Modelling are the same as in Week 1, including the yellow highlight from Week 2. The accuracy benchmark is more than 0.99391 to include in the final experiment.

Data preparation

The Accuracy from replacing Null value with Min, Max, Mean, Mode, and Median. The yellow highlights are include in Final experiment.

Table 7 : The Accuracy from replacing Null value with Min, Max, Mean, Mode, and Median

No	Column name	Min	Max	Mean	Mode	Median
1	Rec_Rank	0.99358	0.99358	0.99358	0.99358	0.99358
2	ast_tov	0.99358	0.99358	0.99340	0.99358	0.99367
3	rimmade	0.99364	0.99364	0.99373	0.99364	0.99391
4	rimmade_rimmiss	0.99391	0.99337	0.99373	0.99391	0.99364
5	midmade	0.99391	0.99391	0.99382	0.99391	0.99346

No	Column name	Min	Max	Mean	Mode	Median
6	midmade_midmiss	0.99391	0.99391	0.99346	0.99391	0.99373
7	rim_ratio	0.99391	0.99364	0.99391	0.99364	0.99355
8	pick	0.99355	0.99453	0.99319	0.99400	0.99301
9	dunks_ratio	0.99453	0.99471	0.99471	0.99471	0.99471
10	dunksmiss_dunksmade	0.99471	0.99462	0.99471	0.99471	0.99471
11	dunksmade	0.99471	0.99444	0.99489	0.99471	0.99471
12	mid_ratio	0.99489	0.99462	0.99462	0.99489	0.99462
13	drtg	0.99462	0.99471	0.99453	0.99453	0.99471

The Accuracy from replacing the Null value with the Iterative Imputer algorithm and could not find the better accuracy number.

Table 8 : The Accuracy from replacing Null value with the Iterative Imputer algorithm

Column Name	Iterative Imputer	Accuracy
dunksmade	Linear Regression	0.99489
dunksmade	Decision Tree Regression	0.99489
dunksmade	Random Forest Regression	0.99489
dunksmade	Gradient Boosting Regression	0.99489
dunksmade	Support Vector Regression	0.99489
dunksmade	K-Nearest Neighbors Regression	0.99489
dunksmade	Neural Network Regression (MLP)	0.99489
dunksmade	Bayesian Ridge Regression	0.99489
dunksmade	Lasso Regression	0.99489
dunksmade	Ridge Regression	0.99489
Rec_Rank	Linear Regression	0.99462
rimmade	Linear Regression	0.99471
ast_tov	Linear Regression	0.99471

The Accuracy from the select number of features is ordered by Information Gain calculation and could not find the better accuracy number.

Table 9 : The Accuracy from the select number of features is ordered by Information Gain

Information Gain Top x	Accuracy
20	0.99435
30	0.99489
35	0.99498
40	0.99498
45	0.99489
50	0.99471

Figure 9 : Confusion Matrix week 3

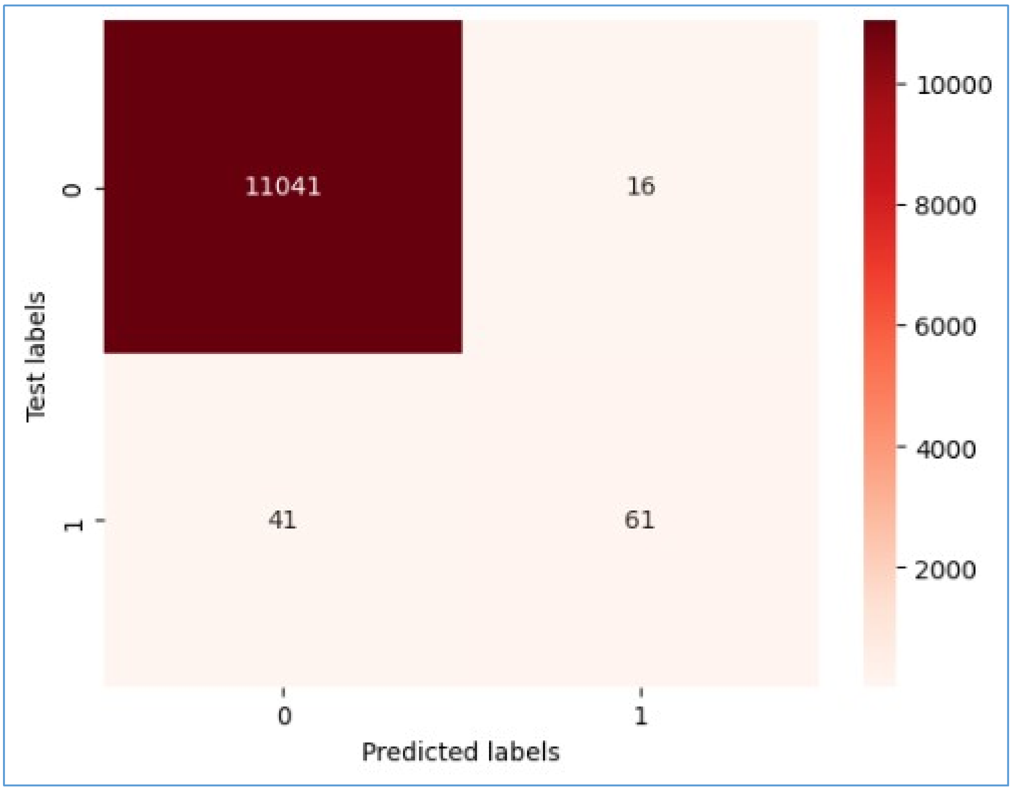
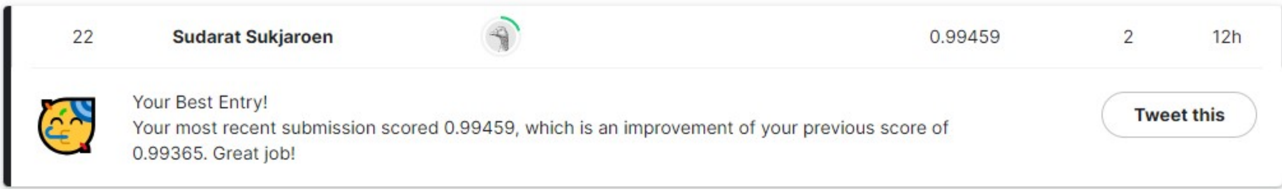


Table 10 : Metric values week 3

Week No.	Accuracy	Precision	Recall	F1 Score
1	0.99349	0.70313	0.45455	0.55215
2	0.99391	0.74286	0.50980	0.60465
3	0.99489	0.79221	0.59804	0.68156

Figure 9 : Keggle ranking week 3



Experiment week 4

Basic data preparation, Feature engineering, and Modelling are the same as in Week 1, including the yellow highlights from Weeks 2 and, 3 and the yellow highlights from week 4 are included in the Final experiment. The accuracy benchmark is more than 0.99489 to include in the final experiment.

Data preparation

Replace Null text data by value in Min, Max, and Average.

Table 11 : The accuracy after replacing Null text data by value in Min, Max, Average

Feature name	Data type	Null count	Empty String Count	Zero Count	Min	Max	Avg/Mean
num	object	4,669	0	415	0.99453	0.99453	0.99453
yr	object	274	0	0	0.99489	0.99489	0.99489
ht	object	80	0	0	0.99489	0.99489	0.99489
team	object	0	0	0			
conf	object	0	0	0			
type	object	0	0	0			
player_id	object	0	0	0			

Table 12 : The accuracy after replacing Zero with Min, Max, Mean, Mode, and Median

Feature name	Data type	Zero Count	Min	Max	Avg/Mean	Mode	Median
TPM	int64	18,222	0.99462	0.99462	0.99462	0.99462	0.99462
TPA	int64	11,709	0.99435	0.99435	0.99435	0.99435	0.99435
FTM	int64	7,474	0.99453	0.99453	0.99453	0.99453	0.99453
FTA	int64	6,272	0.99462	0.99462	0.99462		
twoPM	int64	6,082	0.99471	0.99471	0.99471		
twoPA	int64	3,254	0.99498	0.99498	0.99498		
GP	int64	0					
year	int64	0					
pick	float64	0					
Rec_Rank	float64	0					
dunks_ratio	float64	1,077	0.99471	0.99471	0.99471		
mid_ratio	float64	4,664	0.99471	0.99471	0.99471		
rim_ratio	float64	2,119	0.99471	0.99471	0.99471		
dunksmade	float64	25,789	0.99489	0.99489	0.99489		
dunksmiss_dunksmade	float64	24,712	0.99453	0.99453	0.99453		
midmade	float64	8,271	0.99453	0.99453	0.99453		
rimmade	float64	5,502	0.99480	0.99480	0.99480		
midmade_midmiss	float64	3,607	0.99453	0.99453	0.99453		
rimmade_rimmiss	float64	3,383	0.99471	0.99471	0.99471		
ast_tov	float64	3,258	0.99471	0.99471	0.99471		
blk	float64	14,333	0.99462		0.99462		
stl	float64	7,491			0.99471		
ast	float64	6,286			0.99480		

Feature name	Data type	Zero Count	Min	Max	Avg/Mean	Mode	Median
oreb	float64	6,045			0.99471		
dreb	float64	3,272			0.99489		
pts	float64	3,175			0.99480		
treb	float64	2,508			0.99471		
mp	float64	19			0.99471		
TP_per	float64	18,222	0.99421		0.99498		
blk_per	float64	14,823	0.99480		0.99480		
stl_per	float64	8,000			0.99453		
FT_per	float64	7,474			0.99453		
AST_per	float64	6,831			0.99471		
ftr	float64	6,553			0.99462		
ORB_per	float64	6,495			0.99471		
twoP_per	float64	6,082			0.99489		
eFG	float64	4,605			0.99489		
TO_per	float64	4,576			0.99471		
DRB_per	float64	3,670	0.99507		0.99507		
TS_per	float64	3,661	0.99480		0.99480		
pfr	float64	3,479			0.99480		
Ortg	float64	2,490			0.99480		
usg	float64	1,158			0.99480		
adjoe	float64	68			0.99498		
Min_per	float64	23			0.99471		
porpag	float64	0					

Table 13 : The accuracy after replacing Null with Min, Max, Mean, Mode, and Median

Feature name	Data type	Null count	Zero Count	Min	Max	Avg/Mean	Mode	Median
blk	float64	38	14,333	0.99462		0.99462		
stl	float64	38	7,491			0.99471		
ast	float64	38	6,286			0.99480		
oreb	float64	38	6,045			0.99471		
dreb	float64	38	3,272			0.99489		
pts	float64	38	3,175			0.99480		
treb	float64	38	2,508			0.99471		
mp	float64	38	19			0.99471		

Feature engineering

AdaBoost (Adaptive Boosting): Boosting algorithm that improves classification accuracy by combining weak learners' predictions through weighted voting.

SMOTE (Synthetic Minority Over-sampling Technique): Resampling method for addressing class imbalance by creating synthetic examples of the minority class to balance the dataset.

Table 14 : The accuracy after applying AdoBoost and Smote

No.	Model Name	Accuracy	Precision	Recall	F1
1	AdaBoost	0.99480	0.80556	0.56863	0.66667
2	Smote	0.97957	0.30189	0.94118	0.45714

Local Outlier Factor (LOF) is an anomaly detection algorithm that identifies local outliers by comparing the density of data points to their neighbors, flagging points with significantly lower densities as anomalies.

Isolation Forest is an anomaly detection method that isolates anomalies by creating random decision trees and measuring the number of splits required to isolate an instance, making anomalies stand out as shorter paths in the trees.

Table 15 : Run experiments with other models

No.	Model Name	Parmeters	Accuracy
1	Local outliter Factor (LOF)	n_neighbors=20, contamination=0.1	0.99471
2	Local outliter Factor (LOF)	n_neighbors=5, contamination=0.1	0.99471
3	Local outliter Factor (LOF)	n_neighbors=20, contamination=0.1	0.99471
4	Local outliter Factor (LOF)	n_neighbors=30, contamination=0.1	0.99471
5	Local outliter Factor (LOF)	n_neighbors=5, contamination=0.2	0.99471
6	Local outliter Factor (LOF)	n_neighbors=5, contamination=0.3	0.99471
7	Local outliter Factor (LOF)	n_neighbors=5, contamination=0.4	0.99471
8	Local outliter Factor (LOF)	n_neighbors=5, contamination=0.5	0.99471
9	Isolation Forest	contamination=0.1, random_state=42	0.99471
10	Isolation Forest	contamination=0.01, random_state=42	0.99471
11	Isolation Forest	contamination=0.05, random_state=42	0.99471
12	Isolation Forest	contamination=0.2, random_state=42	0.99471

Table 16 : Run experiments with other models

No.	Model	Accuracy	Precision	Recall	F1 Score
1	Decision Tree	0.99453	0.74699	0.60784	0.67027
2	Extra Tree	0.99480	0.89286	0.49020	0.63291
3	Random Forest	0.99489	0.83582	0.54902	0.66272
4	Naive Bayes	0.87257	0.06579	0.98039	0.98039
5	k-Nearest Neighbors	0.99086	0.50000	0.00980	0.01923

Figure 10 : Confusion Matrix week 4

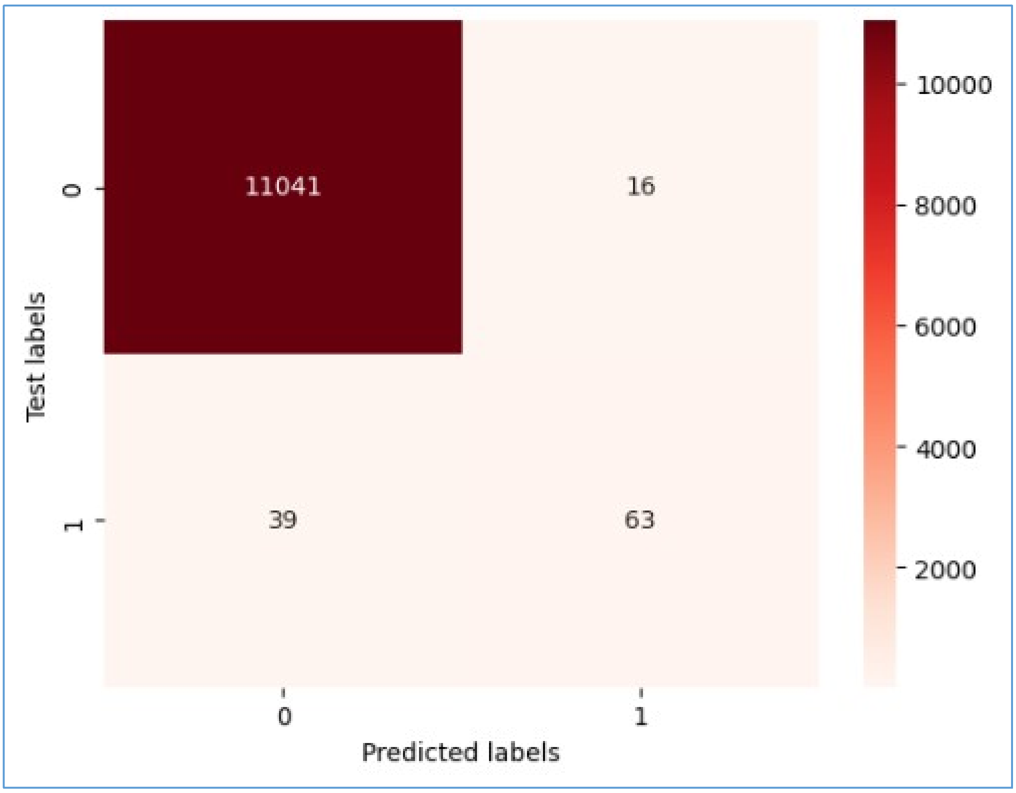
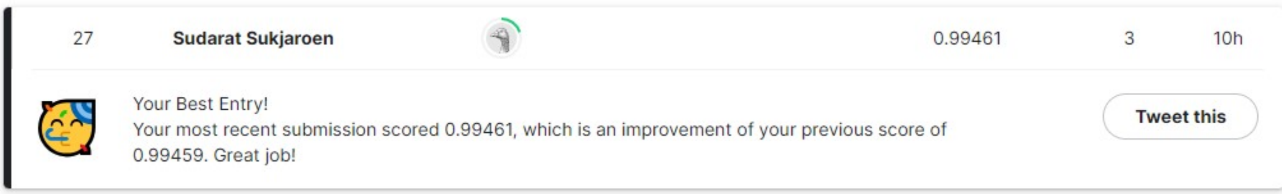


Table 17 : Metric values week 4

Week No.	Accuracy	Precision	Recall	F1 Score
1	0.99349	0.70313	0.45455	0.55215
2	0.99391	0.74286	0.50980	0.60465
3	0.99489	0.79221	0.59804	0.68156
4	0.99507	0.79747	0.61765	0.69613

Figure 11 : Keggle ranking week 4



5. Evaluation

5.1 Evaluate results

For the Kaggle competition, I am not satisfied with the result too much. In the first week, I expected to be in the Top 10, but finally, I am in the Top 30.

5.2 Review process

In the first week, I found many null and missing values, then I focused on them and tried to improve, but the accuracy value did not increase significantly. If I go with Logistic regression, I should concentrate and research more on techniques like feature scaling, regularization, and model evaluation metrics such as the likelihood ratio test, AIC, BIC, and ROC-AUC to assess and improve the model's performance and interpretability.

6. Deployment

6.1 Plan deployment

Python code developed on Google Colab and kept in Github repository https://github.com/sudarat-pom/AdvanceML_AT1. It is ready to adjust on deploy in the production environment.

6.2 Review project

<u>Problem</u>
Actually, I enjoyed this assignment very much. It was good to compete with my classmates. I am passionate about improving my accuracy value, but It was harder than I thought in the first week. I did almost one hundred experiments, and the value increased to only 0.0001.

<u>Solution</u>
I did all I planned to experiment, but it only increased a little. I should research how to improve specific models, such as Logistic regression.

7. Suggestions / Recommendations

If you found the best model for your data, not only data cleansing, you should research more in that model to make the best performance on it.

8. Discussion of ethics/privacy issues

- Privacy: Anonymize and protect player data to respect privacy rights.
- Consent: Ensure informed consent for data usage.
- Bias Mitigation: Address and mitigate biases in data and models.
- Fairness: Monitor and correct for unfair predictions.
- Transparency: Make model decisions clear.
- Explainability: Provide interpretable model explanations.
- Monitoring: Continuously assess model performance and fairness.
- Data Retention: Define secure data retention policies.
- Legal Compliance: Comply with data protection laws.
- Ethical Guidelines: Follow ethical best practices.
- Public Transparency: Share model details with transparency.
- Feedback: Allow stakeholders to provide feedback on predictions.