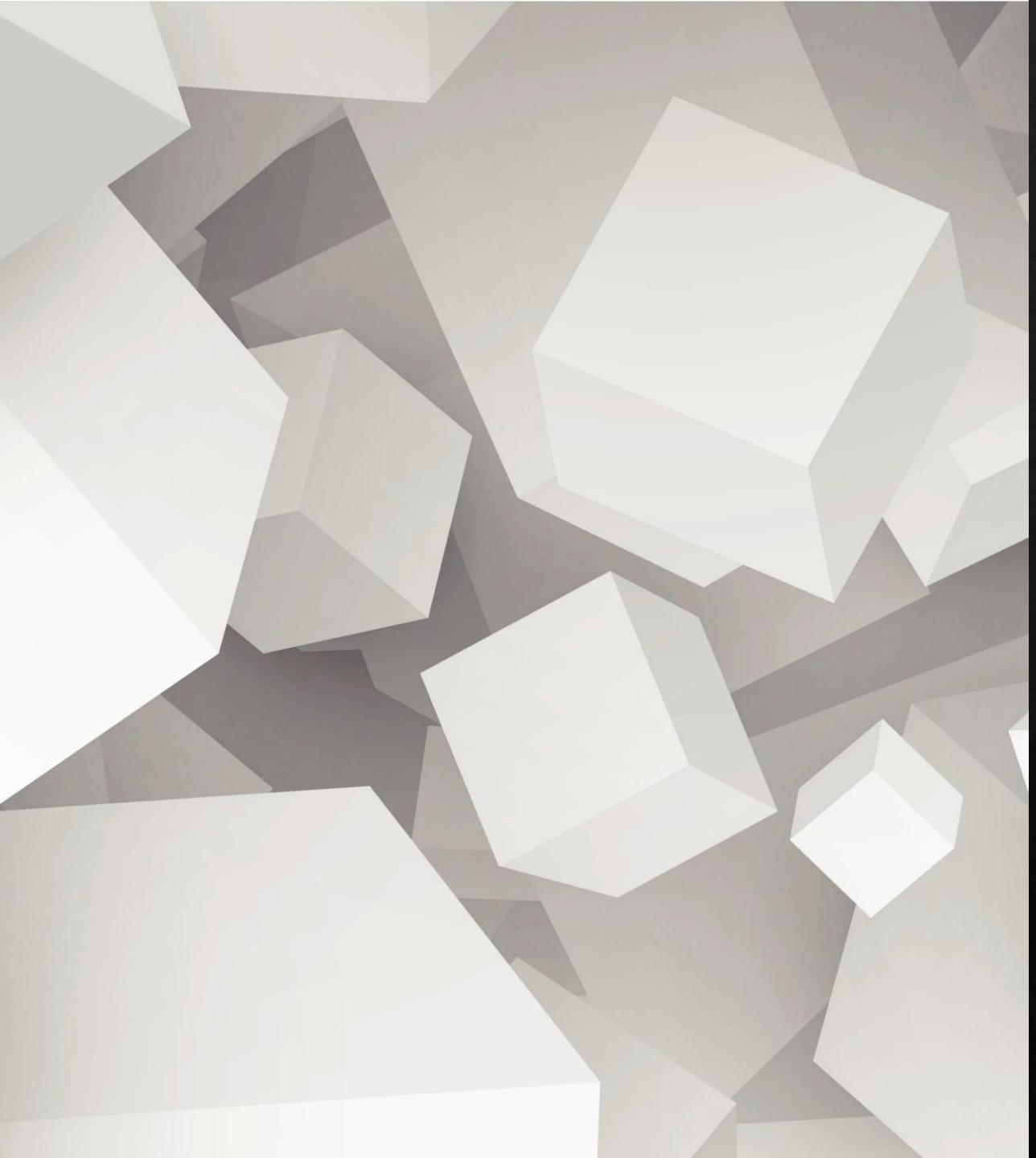
The Insight of Entropy in Soccer Matches

Jackson Nahom



Agenda

What is Entropy?

The Data

Application of Entropy

Entropy Results

Machine Learning Models

Conclusion

What is Entropy?

- Entropy is the measure of uncertainty associated with random variables. (Shannon 1948)
- Shannon Entropy
- $H(x) = -\sum_x p(x) log_2 p(x)$

The Data

- Event driven data for soccer matches
- Events: shots, blocks, passes, interceptions, dribbles, foul committed, clearance, etc.
- To use the data we need to weight each event in a way that all the probabilities add up to 0 for each game
- Some Events are positive for a team and some are negative

Event	Weight
Pass	0.05
Shot	4
Foul Committed	0.5
Interception	1.5
Dispossessed	1
Block	0.5
Ball Receipt	0.2

StatsBomb

How the weighting is done

- Take all the events that occur in the game and group them by how many of each event occurs
- Multiply the grouping by the weighting to achieve a value for the event
- Divide each individual value by the total value the game for a probability for each event
- Finally Divide the event probability by the number of occurrences (Individual Probability)

	type	id	weight	value	total_prob	indv_prob
0	Bad Behaviour	1	1.00	1.00	0.002292	0.002292
1	Ball Receipt*	1058	0.08	84.64	0.193964	0.000183
2	Ball Recovery	89	0.50	44.50	0.101978	0.001146
3	Block	32	0.20	6.40	0.014666	0.000458
4	Carry	890	0.05	44.50	0.101978	0.000115
5	Clearance	37	0.00	0.00	0.000000	0.000000
6	Dispossessed	21	1.00	21.00	0.048124	0.002292
7	Dribble	24	0.00	0.00	0.000000	0.000000
8	Dribbled Past	14	2.00	28.00	0.064166	0.004583
9	Duel	53	0.80	42.40	0.097165	0.001833
10	Error	1	1.00	1.00	0.002292	0.002292
11	Foul Committed	23	0.50	11.50	0.026354	0.001146
12	Foul Won	21	0.10	2.10	0.004812	0.000229
13	Goal Keeper	34	0.00	0.00	0.000000	0.000000
14	Half End	4	0.00	0.00	0.000000	0.000000
15	Half Start	4	0.00	0.00	0.000000	0.000000
16	Interception	24	1.00	24.00	0.054999	0.002292
17	Miscontrol	17	0.10	1.70	0.003896	0.000229
18	Pass	1163	0.01	11.63	0.026652	0.000023
19	Pressure	212	0.00	0.00	0.000000	0.000000
20	Shot	28	4.00	112.00	0.256663	0.009167
21	Substitution	6	0.00	0.00	0.000000	0.000000
22	Tactical Shift	4	0.00	0.00	0.000000	0.000000

Weighting Code

- Teams: Barcelona v Deportivo Alavés
- Value Total: 436.37
- Number of Events: 3760

```
df_prob = df_drop.groupby(['type']).count()['id'].reset_index()
df_prob['weight'] = df_prob['type'].map(event_weight_dict)
df_prob['value'] = df_prob['id']*df_prob['weight']
total = df_prob['value'].sum()
print(total)
print(df_prob['id'].sum())
df_prob['total_prob'] = df_prob['value']/total
print(df_prob['total_prob'].sum())
df_prob['indv_prob'] = df_prob['total_prob']/df_prob['id']
df_prob
```

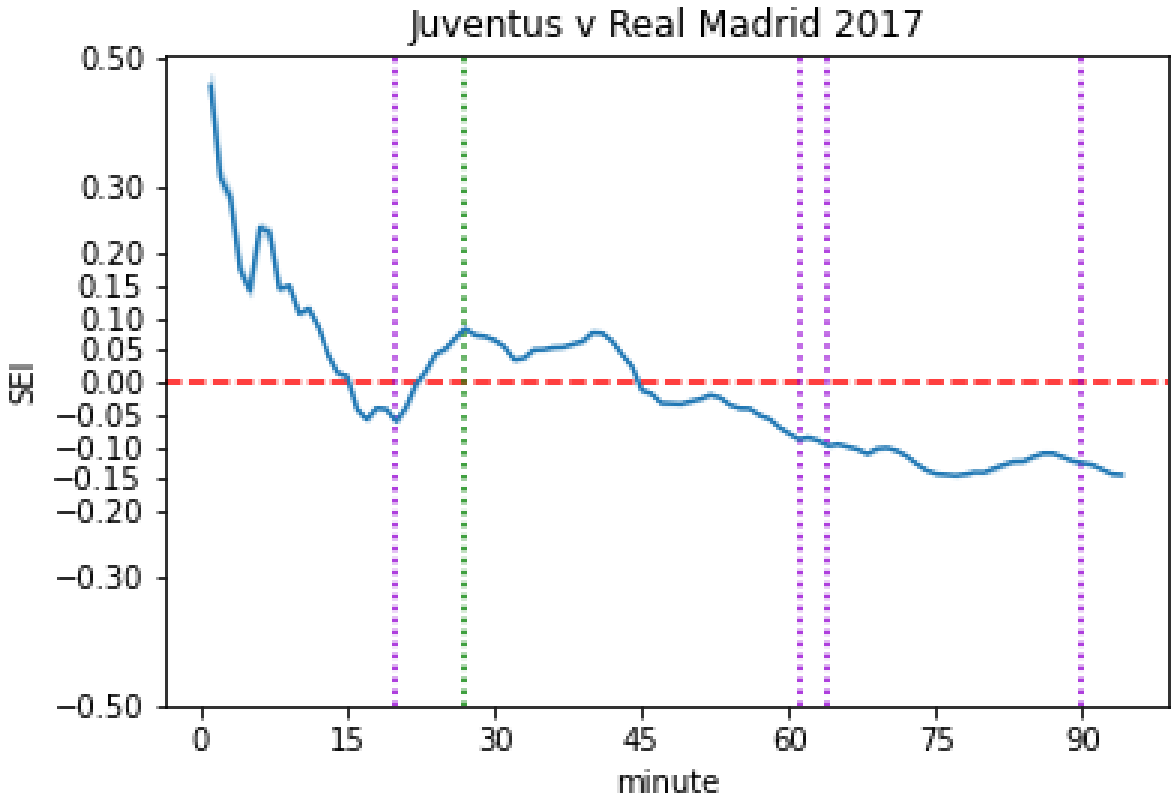
Application of Entropy

- Positive Events between teammates (T1_T1 or T2_T2)
- Positive Events for one team and Negative Events for the other (T1_T2 or T2_T1)
- This creates a matrix $\begin{bmatrix} T1\_T1 & T1\_T2 \\ T2\_T1 & T2\_T2 \end{bmatrix}$
- We use the Shannon Entropy Equation based on the individual probability for each event for the element of the Matrix
- Then T1 Entropy Segment = $T1\_T1 + T2\_T1 - T1\_T2$

  T2 Entropy Segment = $T2\_T2 + T1\_T2 - T2\_T1$
- Shannon Entropy Index = T1 Entropy Segment – T2 Entropy Segment
- Positive Shannon Entropy Index (SEI) would indicate a win for the home team

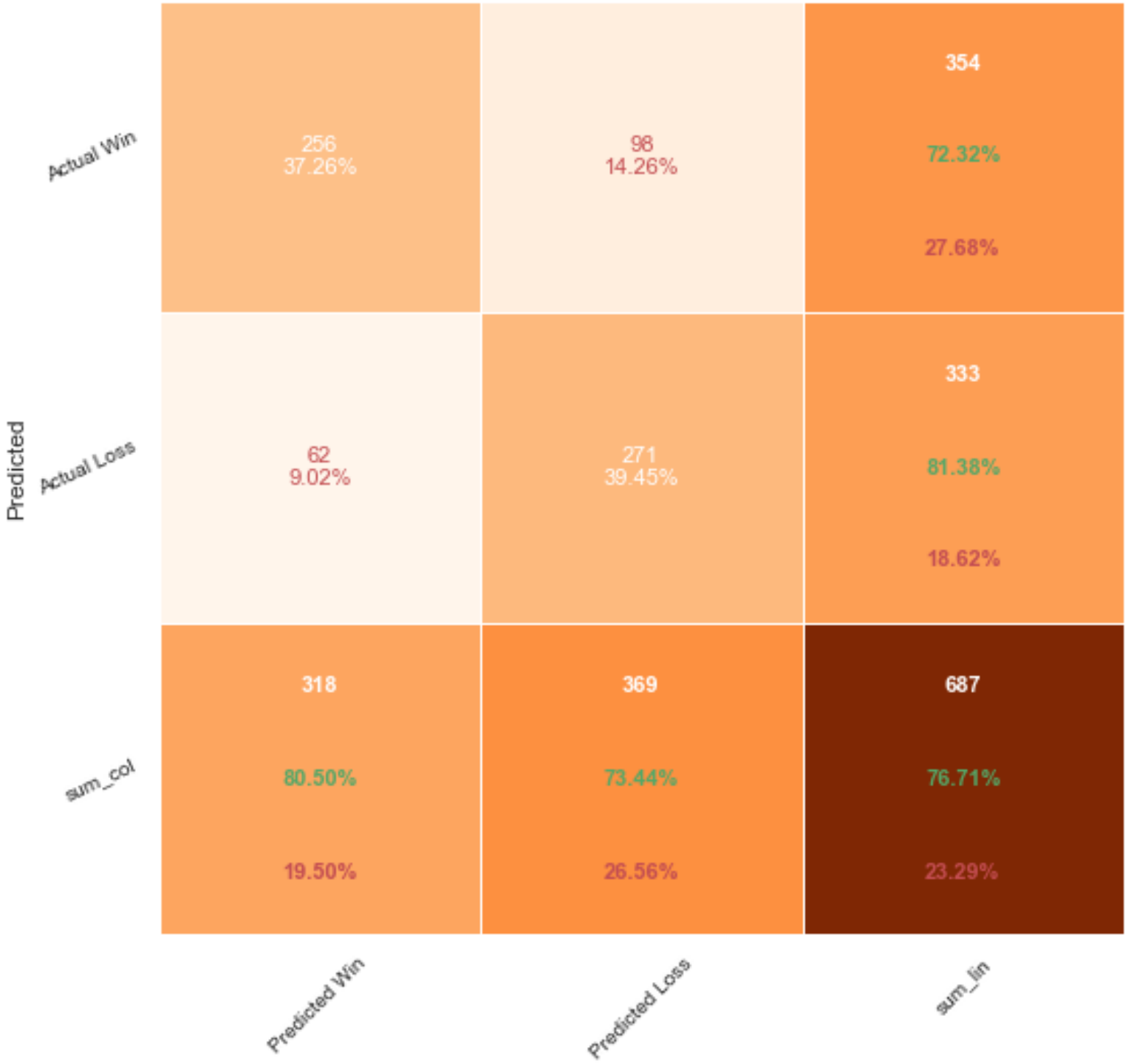
Entropy Through out a Match

- Green is Juventus Goals
- Purple is Real Madrid Goals
- SEI correctly predicted a Real Madrid Win
- Entropy application:
  $current\ Entropy = Last\ Row\ + (-(p(x)log_2 p(x)))$

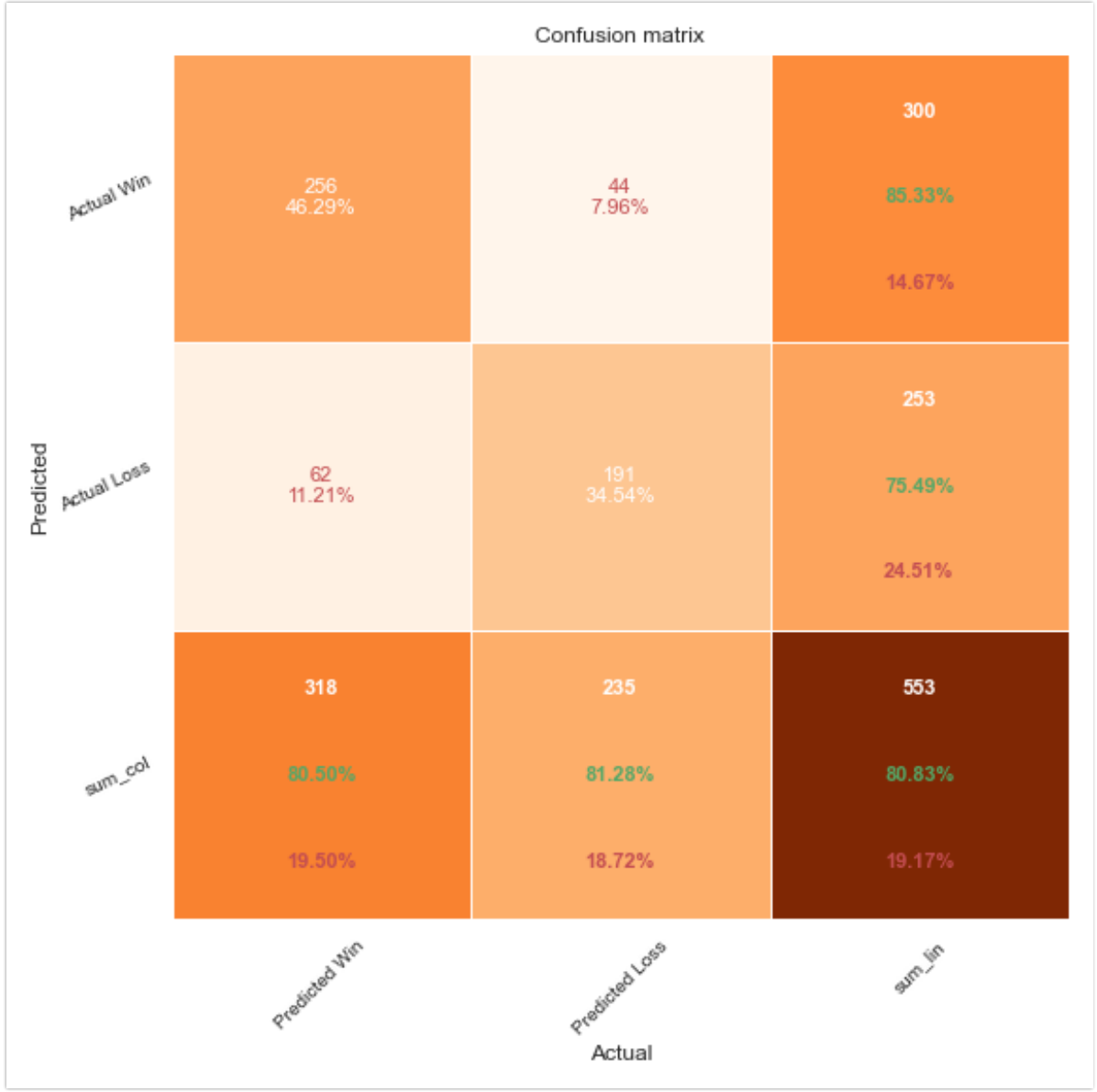


Entropy Results

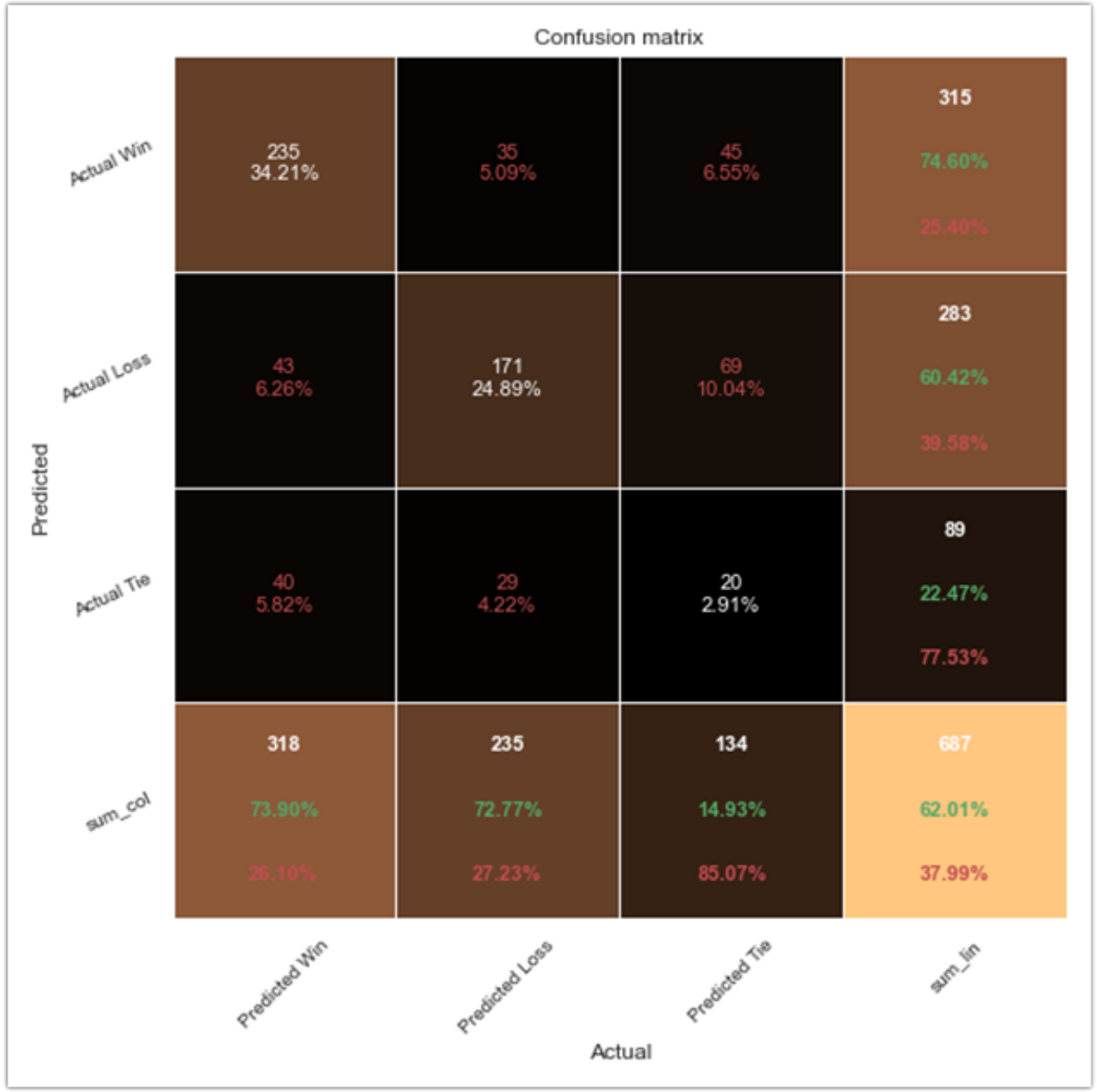
- When we predict a win we are right 80.5% of the time
- When we predict not a win we are right 73.44% of the time
- Total SEI is correct 76.71% of the time
- This includes matches with ties



Removing Ties

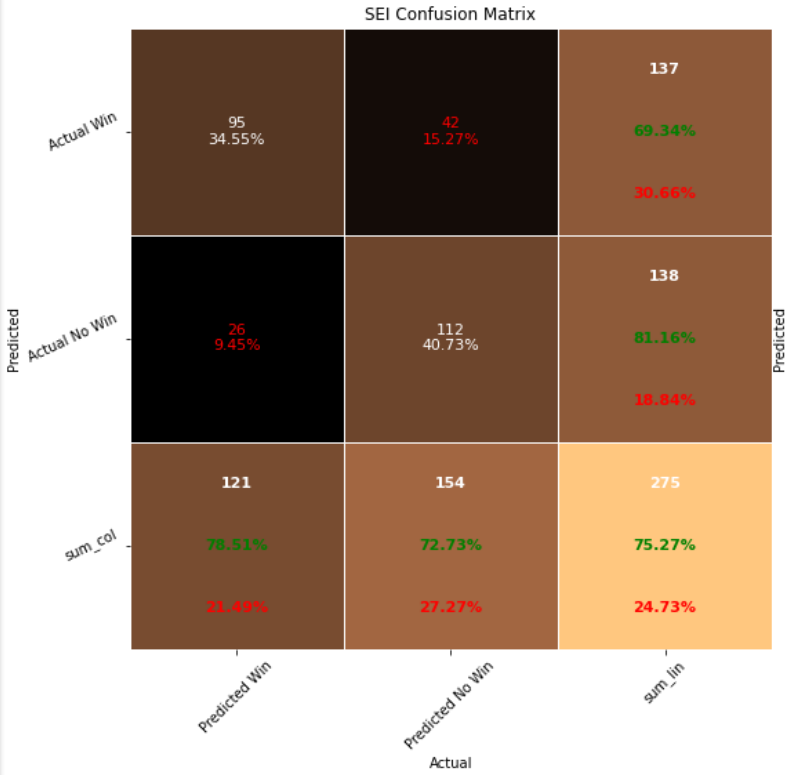


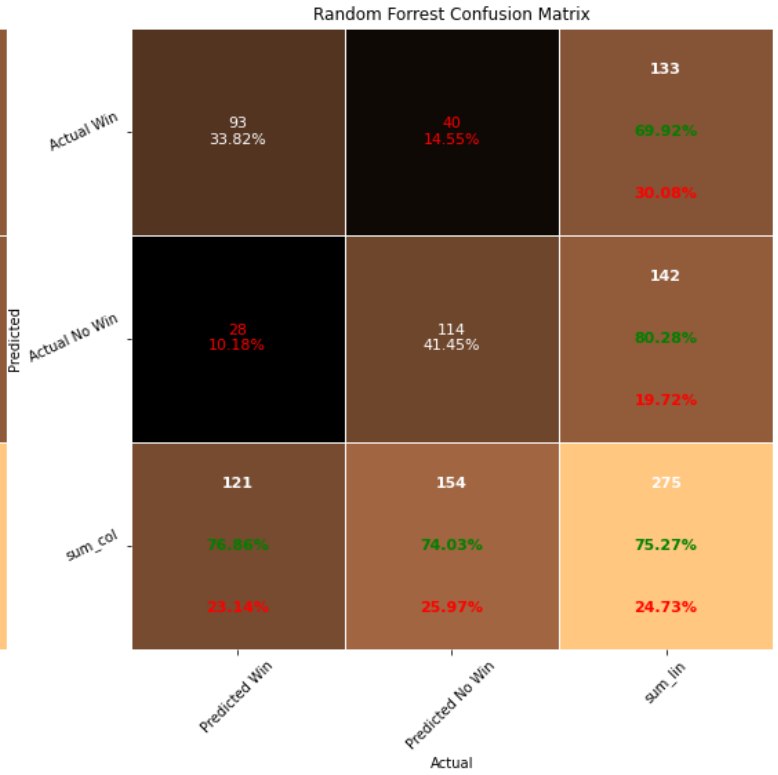
Attempt to Predict Ties



Random Forrest Classifier

- Goal was to see if ML models could make a better decision than my SEI
- Performed almost Identically as SEI on the same test set
- The Features used were: T1_T1, T1_T2, T2_T2, and T2_T1 Entropies





Other Models

- Project only looked at 637 Matches so not a lot data when compacted to matches
- Best Performing Model according to AUC was MLP

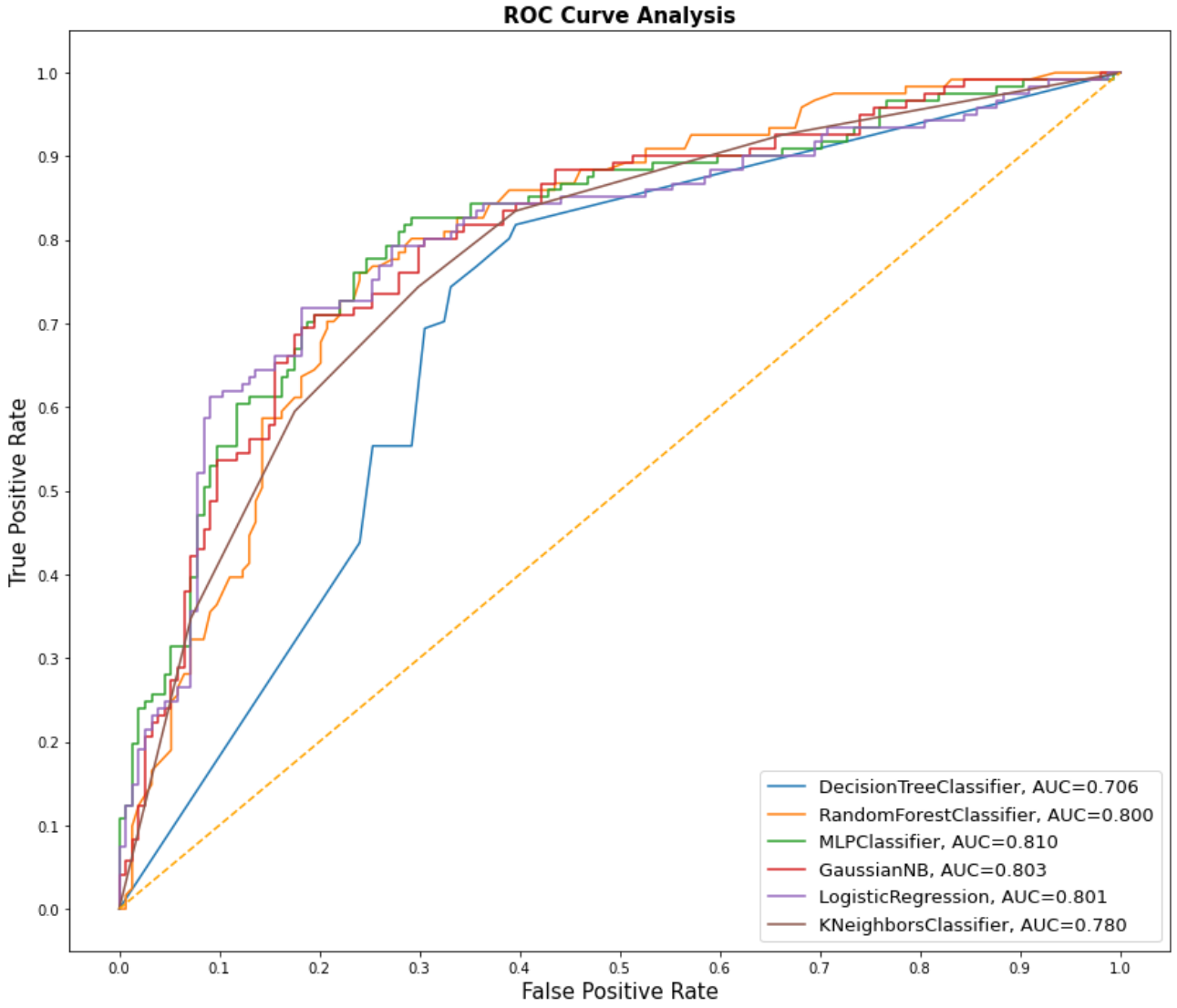
```
---- statistics for MLPClassifier ----

              precision    recall  f1-score   support

           0       0.80      0.76      0.78       154
           1       0.71      0.76      0.74       121

    accuracy                           0.76       275
   macro avg       0.76      0.76      0.76       275
weighted avg       0.76      0.76      0.76       275

Model log loss: 0.5318158829769136
```



Simple Sklearn Code

```
from sklearn.model_selection import train_test_split
# use index-based sampling since we have time series data
train, test = train_test_split(df, test_size=0.4, shuffle=True)
```

```
pred_vars = ['t1_t1_entropy','t2_t2_entropy', 't1_t2_entropy', 't2_t1_entropy']
y_var = 'home_win'
```

```
dtree = tree.DecisionTreeClassifier(criterion='entropy', max_depth=10)
dtree.fit(train[pred_vars], train[y_var])
```

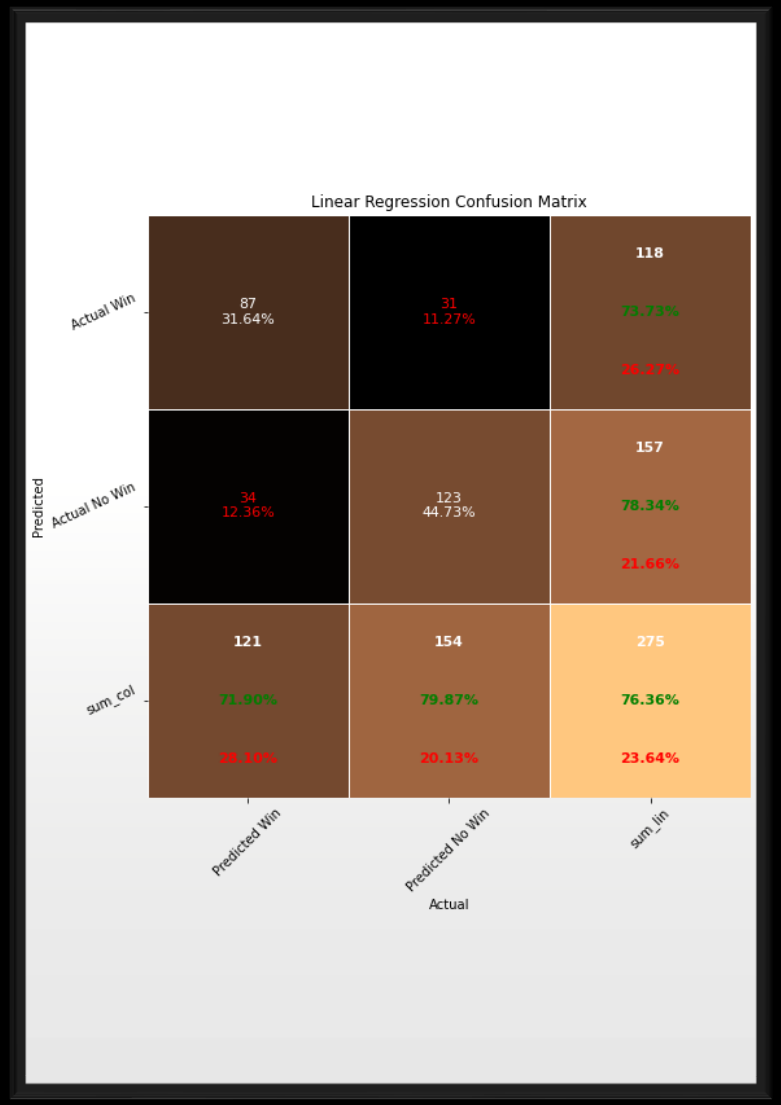
```
rf = ensemble.RandomForestClassifier(n_estimators=200)
rf.fit(train[pred_vars], train[y_var])
```

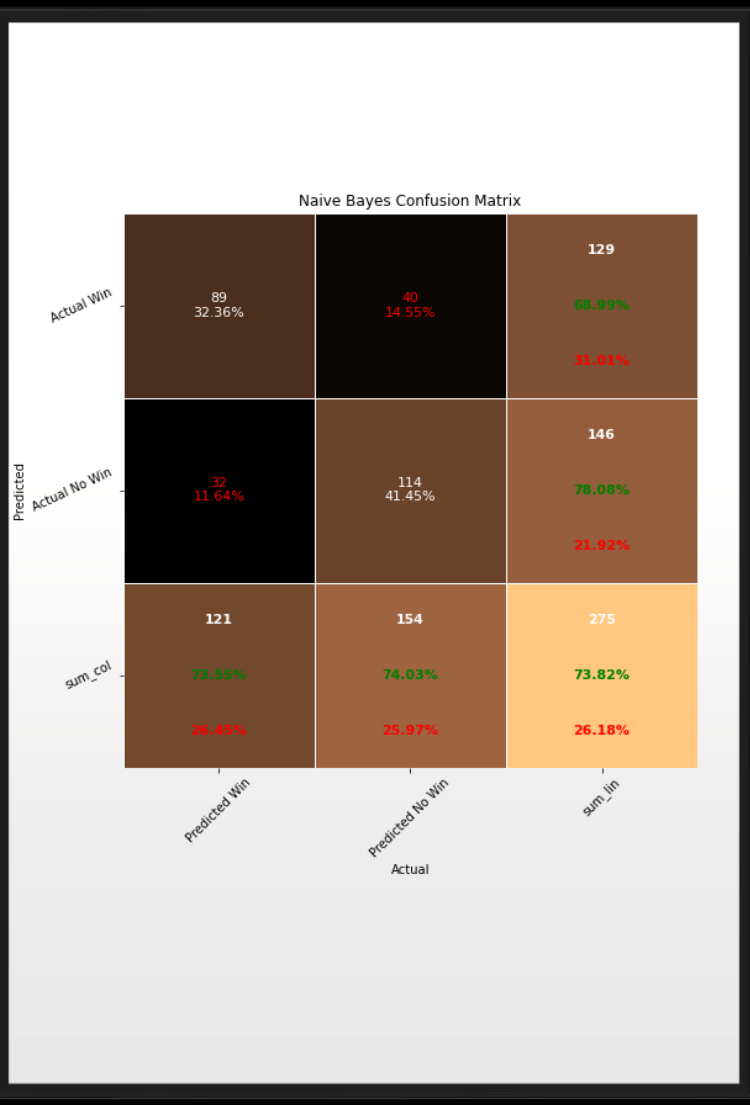
```
mlp = MLPClassifier(hidden_layer_sizes=(20, 20), max_iter=500, n_iter_no_change=20)
mlp.fit(train[pred_vars], train[y_var])
```

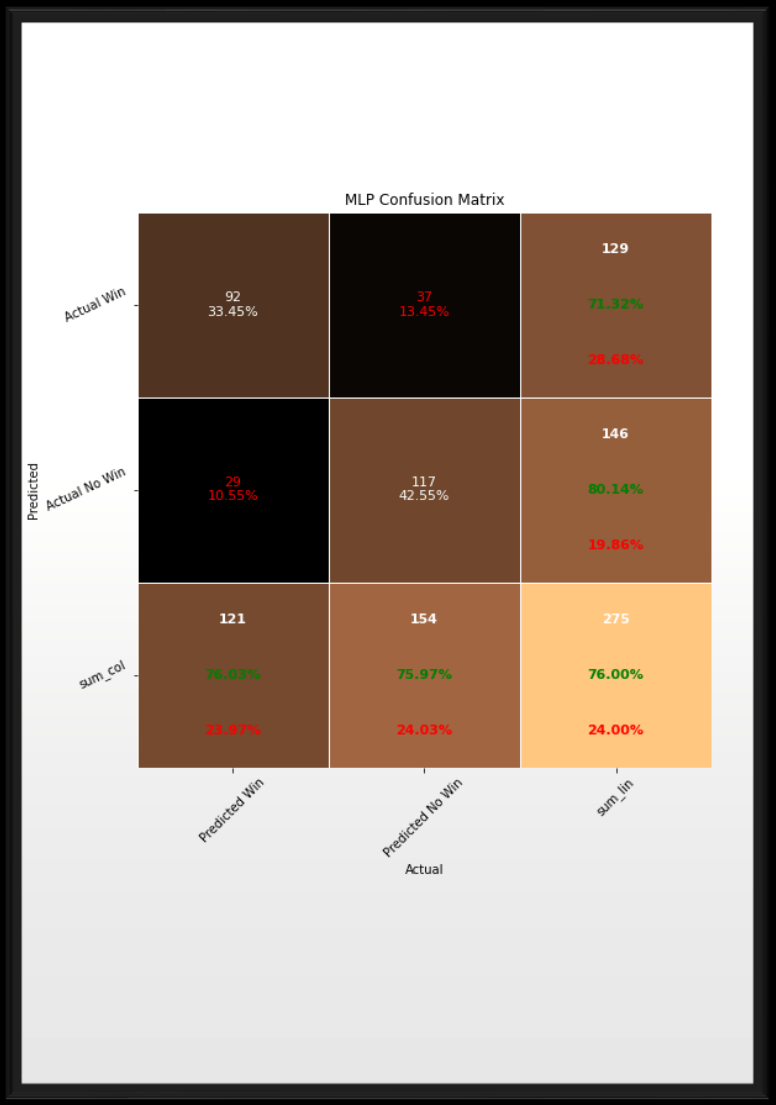
```
nb = GaussianNB()
nb.fit(train[pred_vars], train[y_var])
```

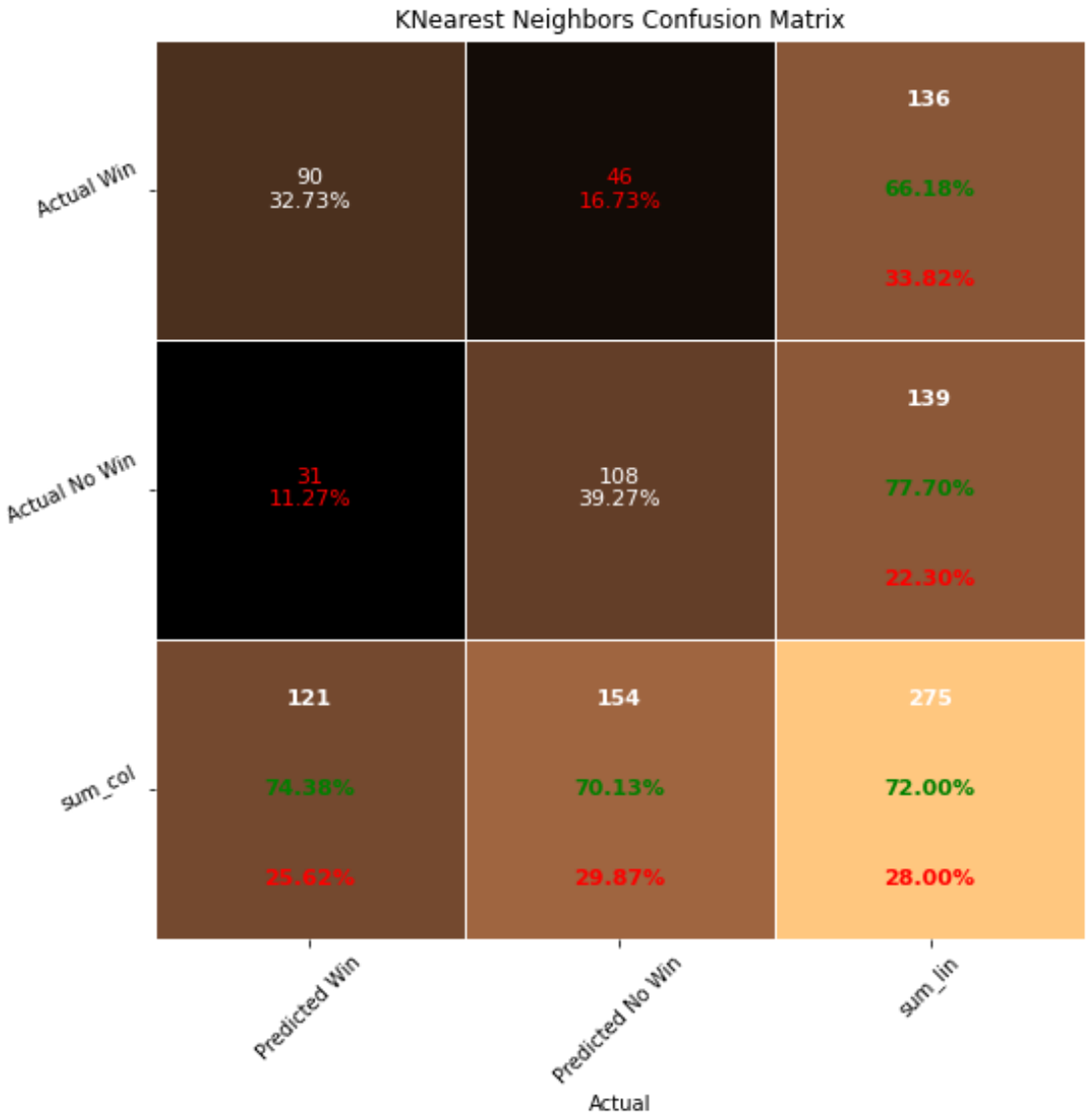
```
lr = LogisticRegression()
lr.fit(train[pred_vars], train[y_var])
```

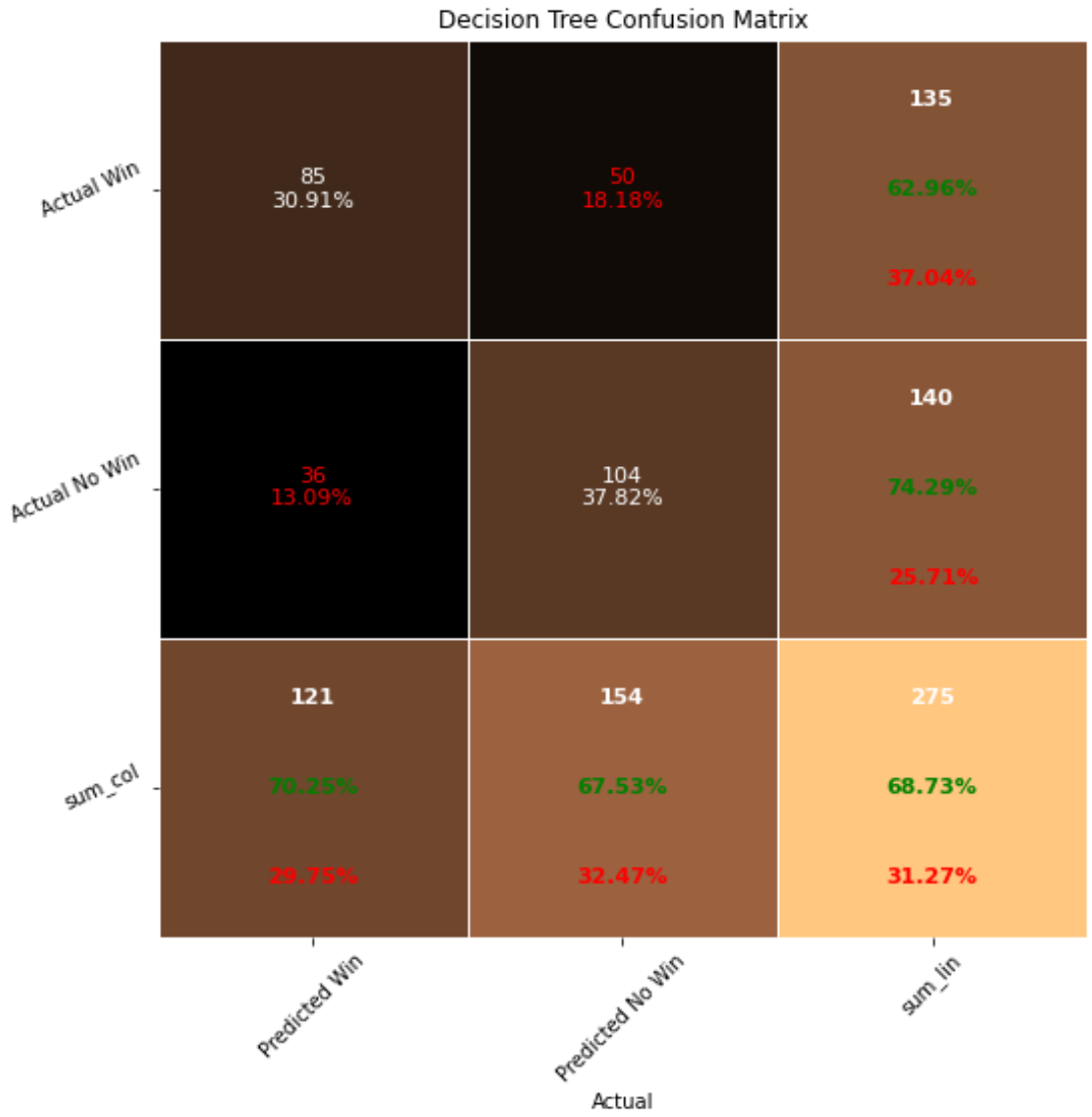
```
knr = KNeighborsClassifier()
knr.fit(train[pred_vars], train[y_var])
```


Linear Regression Confusion Matrix
Actual Win
Actual No Win
sum_col
Predicted
87
31.64%
31
11.27%
118
73.73%
26.27%
34
12.36%
123
44.73%
157
78.34%
21.66%
121
71.90%
28.10%
154
79.87%
20.13%
275
76.36%
23.64%
Predicted Win
Predicted No Win
sum_lin
Actual


Naive Bayes Confusion Matrix
Actual Win
Actual No Win
sum_col
Predicted
89
32.36%
40
14.55%
129
68.99%
31.01%
32
11.64%
114
41.45%
146
78.08%
21.92%
121
73.55%
26.45%
154
74.03%
25.97%
275
73.82%
26.18%
Predicted Win
Predicted No Win
sum_lin
Actual


MLP Confusion Matrix
Actual Win
Actual No Win
sum_col
Predicted
92
33.45%
37
13.45%
129
71.32%
28.68%
29
10.55%
117
42.55%
146
80.14%
19.86%
121
76.03%
23.97%
154
75.97%
24.03%
275
76.00%
24.00%
Predicted Win
Predicted No Win
sum_lin
Actual


KNearest Neighbors Confusion Matrix
Actual Win
Actual No Win
sum_col
90
32.73%
46
16.73%
136
66.18%
33.82%
31
11.27%
108
39.27%
139
77.70%
22.30%
121
74.38%
25.62%
154
70.13%
29.87%
275
72.00%
28.00%
Predicted Win
Predicted No Win
sum_lin
Actual


Decision Tree Confusion Matrix
Actual Win
Actual No Win
sum_col
85
30.91%
50
18.18%
135
62.96%
37.04%
36
13.09%
104
37.82%
140
74.29%
25.71%
121
70.25%
29.75%
154
67.53%
32.47%
275
68.73%
31.27%
Predicted Win
Predicted No Win
sum_lin
Actual

Improvements

- Larger Data Set
- For the models could have given them some more information, for example the number of times each event happened in a game
- Could come up with other Index Types such as KL Distance, Kolmogorov Complexity, or a total activity Index where weighting didn't matter

Conclusion

- Successful
- My SEI performed just as well as the ML model with the same information
- A lot can be learned just by using the simple Shannon Entropy Equation

Questions?

Sources

- C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

- S. Kusmakar, S. Shelyag, Y. Zhu, D. Dwyer, P. Gastin, and M. Angelova, “Machine learning enabled team performance analysis in the dynamical environment of soccer,” *IEEE Access*, vol. 8, pp. 90266–90279, 2020.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9085411

StatsBomb: https://statsbomb.com/what-we-do/hub/free-data/

GitHub Repo

- https://github.com/nahomjd/open-data/tree/working

Appendix

---- statistics for KNeighborsClassifier ----

	precision	recall	f1-score	support
0	0.75	0.75	0.75	146
1	0.72	0.72	0.72	129
accuracy			0.73	275
macro avg	0.73	0.73	0.73	275
weighted avg	0.73	0.73	0.73	275

Model log loss: 3.1049925404774976

---- statistics for LogisticRegression ----

	precision	recall	f1-score	support
0	0.80	0.77	0.78	146
1	0.75	0.78	0.77	129
accuracy			0.77	275
macro avg	0.77	0.78	0.77	275
weighted avg	0.78	0.77	0.77	275

Model log loss: 0.5479686361291668

---- statistics for GaussianNB ----

	precision	recall	f1-score	support
0	0.79	0.73	0.75	146
1	0.71	0.78	0.74	129
accuracy			0.75	275
macro avg	0.75	0.75	0.75	275
weighted avg	0.75	0.75	0.75	275

Model log loss: 0.8200295525276919

---- statistics for RandomForestClassifier ----

	precision	recall	f1-score	support
0	0.77	0.75	0.76	146
1	0.72	0.74	0.73	129
accuracy			0.75	275
macro avg	0.74	0.75	0.74	275
weighted avg	0.75	0.75	0.75	275

Model log loss: 0.5874722089800447

---- statistics for DecisionTreeClassifier ----

	precision	recall	f1-score	support
0	0.70	0.73	0.72	146
1	0.68	0.64	0.66	129
accuracy			0.69	275
macro avg	0.69	0.69	0.69	275
weighted avg	0.69	0.69	0.69	275

Model log loss: 8.791196256173054

Event	Weight	Event	Weight	Event	Weight	Event	Weight
Starting XI	0	Block	0.5	Dribbled Past	2	50/50	0
Half Start	0	Interception	1.5	Bad Behaviour	1	Camera On	0
Pass	0.05	Shot	4	Half End	0	Offside	0
Ball Receipt	0.2	Goal Keeper	0	Substitution	0	Own Goal Against	0
Carry	0.1	Clearance	0	Tactical Shift	0	Own Goal For	0
Duel	1	Dispossessed	1	Error	1	Player On	0
Ball Recovery	0.5	Dribble	0	Shield	1	Player Off	0
Pressure	0	Foul Committed	0.5	Referee Ball-Drop	0	Camera On	0
Miscontrol	0.1	Foul Won	0.8	Injury Stoppage	0	Camera off	0