2023

HACETTEPE
University

VBM 683
Machine Learning

BUILDING AN ML MODEL TO PREDICT CORNER COUNT AFTER AN EXACT MINUTES OF MATCH BY USING IN-GAME FOOTBALL MATCH DATA

OGÜN ŞERIF ONARGAN
N22137400

1. Introduction

Soccer is one of the most popular sport which has defended its popularity for many decades. One of the reason of this popularity is occurrence rate of capriciousness events. Because of that reason, soccer has great betting market size.

The betting market offers odds for football events by the help of widely soccer data and its Bayes' theorem applications. Also, odds of live betting lean on this method, because of odds' continuity, scalability and generalizability. This situation can be an opportunity to increase winning rate by using in-game data for a specific match. The main aim of the project is to predict number of corner occurs in specific time by using in-game data before the specific time.

Dataset is taken from Kaggle which is created by Wyscout.

https://www.kaggle.com/datasets/aleespinosa/soccer-match-event-dataset

Colab Notebook can be access with,

EDA Notebook:

https://colab.research.google.com/drive/1jaLZUkwUgLiBFsEn2sdh56bXF3jvVS9n?usp=sharing

ML Notebook:

https://colab.research.google.com/drive/1IH0ahj0YTdsN0bmx2LN4TP67WiaBuErR?usp=sharing

2. Determination of Project Goal

Assume a coin is flipped. Its ratio of outcomes approaches 0.5 for each possibility while sample size goes infinity. Betting market use decimal odds which is calculated below,

$$\frac{1}{0.5} = 2.00$$

If a betting company declare odds of flipped coin, it cannot ensure its earning. Therefore, it takes its commission from probability. For example,

Rate of commission: 5%,
Total Outcome: 105%,
For each outcome: 0.525

$$\frac{1}{0.525} = 1.90$$

Let's do the same calculation for a real odd,

Göztepe – Adana Min:73 – Current # of Corner: 7		
	Under	Over
Corner: 9.5	1.87	1.87
Corner 10.5	1.53	2.40

Table (1): An Example of Odds

For first odd: $\frac{1}{1.87} * 2 = 1.0695$

For second odd: $\frac{1}{1.53} + \frac{1}{2.4} = 1.070$

It can be easily assumed that the commission is 7%, so the project goal can be determined as if a ML model predict the outcome more than its probability in dataset plus 3.5%, balance has positive total at end of the day.

3. Explanatory Data Analysis

a. Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2462726 entries, 0 to 2462725
Data columns (total 89 columns):
 #    Column              Dtype
---   ------              -----
 0    Unnamed: 0          int64
 1    game_id-0           int64
 2    period_id-0         int64
 3    time_seconds-0      float64
 4    team_id-0           int64
 5    player_id-0         int64
 6    start_x-0           float64
 7    start_y-0           float64
 8    end_x-0             float64
 9    end_y-0             float64
 10   bodypart_id-0       int64
 11   type_id-0           int64
 12   result_id-0         int64
 13   type_name-0         object
 14   result_name-0       object
 15   bodypart_name-0     object
 16   time_played-0       float64
 17   game_id-1           float64
 18   period_id-1         float64
 19   time_seconds-1      float64
 20   team_id-1           float64
 21   player_id-1         float64
 22   start_x-1           float64
 23   start_y-1           float64
 24   end_x-1             float64
 25   end_y-1             float64
 26   bodypart_id-1       float64
 27   type_id-1           float64
 28   result_id-1         float64
 29   type_name-1         object
 30   result_name-1       object
 31   bodypart_name-1     object
 32   time_played-1       float64
 33   game_id-2           float64
 34   period_id-2         float64
 35   time_seconds-2      float64
 36   team_id-2           float64
 37   player_id-2         float64
 38   start_x-2           float64
 39   start_y-2           float64
 40   end_x-2             float64
 41   end_y-2             float64
 42   bodypart_id-2       float64
 43   type_id-2           float64
 44   result_id-2         float64
 45   type_name-2         object
 46   result_name-2       object
 47   bodypart_name-2     object
 48   time_played-2       float64
```

Table (2): Info of Dataset's Columns

Game_Id, period_id, time_seconds, start_x, start_y, end_x, end_y, result_id and type_name are used to analyze and create final dataset during the project.

Raw Dataset consists of 17x3 columns which are replication of 2 previous event, labeled as 0,1,2. The dataset has 2.462.725 rows.

- Game_ID has 1.941 unique elements. Final dataset will be created a row for each game.
- Period_id determines which half of a game.
- Time_Seconds determines time that an event occurs.
- Start_x, start_y, end_x, end_y determines coordinates of events that start and finish. The dataset always makes defending teams' keep's x-axis zero.
- Result_id determines an event success or fail as binary.
- Type_name determines type of events. It consists of pass, cross, clearance, throw_in, dribble, foul, freekick_crossed, freekick_short, interception, goalkick, take_on, corner_crossed, shot, keeper_save, tackle, corner_short, shot_freekick, shot_penalty, bad_touch.

	Unnamed: 0	game_id-0	period_id-0	time_seconds-0	team_id-0	player_id-0	start_x-0	start_y-0	end_x-0	end_y-0	bodypart_id-0	type_id-0	result_id-0	type_name-0	result_name-0	bodypart_name-0	time_played-0
0	0	2500089	1	2.763597	1659	9637	52.50	34.00	63.00	30.60	0	0	1	pass	success	foot	2.763597
1	1	2500089	1	4.761353	1659	8351	63.00	30.60	64.05	10.20	0	0	1	pass	success	foot	4.761353
2	2	2500089	1	5.533097	1659	9285	64.05	10.20	72.45	20.40	0	0	1	pass	success	foot	5.533097
3	3	2500089	1	7.707561	1659	239411	72.45	20.40	35.70	19.04	0	0	1	pass	success	foot	7.707561
4	4	2500089	1	11.614943	1659	9637	35.70	19.04	30.45	12.24	0	0	1	pass	success	foot	11.614943

Table (3): First 5 columns of Dataset

b. Data Analysis – What causes corner?

During data analysis phase, events are analyzed by using x-axis intervals and x-axis intervals. Events correlation and ratio plots are created. In correlation plot, getting highest correlations from intervals with same trend is discovered.

```
type_name-1
interception          7458
clearance             5219
pass                  3163
keeper_save           2323
tackle                 458
take_on                348
cross                   98
corner_crossed          71
shot                    52
dribble                 35
freekick_crossed        33
throw_in                27
freekick_short          15
corner_short             5
goalkick                 5
shot_freekick            2
Name: Unnamed: 0, dtype: int64
```

Table (4): Event Counts Causes Corner

Interception

Total Count : 133.174

Pre-Corner Count: 7.458

Rate : 5.6%

As it seen in first graph, plot of ratio of interception cause corner and total corner shows us probability of corner occurrence is decreasing while x increases.

Second plot shows us correlation between number of interception and corner occurrence in t>67.5min.

Correlations:

Interception 40-46 (Corr: -0.077): It is valid, because it defines good defensive plays that obstruct opponents approach to keep.

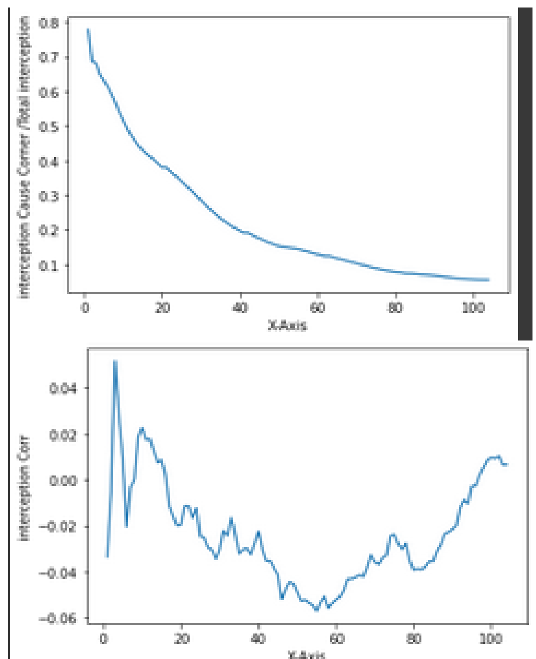
Interception 83-100 (Corr: 0.088): It is valid, because it defines good defensive plays that rapidly take the ball from opponents' defenders.

Lastly, events cause interceptions that causes corners are listed. Cross, shot and pass take great part of table.

Events cause corner are listed in Table (4). According to it, interception, clearance, pass and keeper save are main reason of corners. However, 3 of them are defending events, so they must be investigated more deeply to identify their previous events.

First strategy is decreasing total number of an event occurrence while pre-corner count of an event stays same by setting x-axis interval.

If offensive movement is done before corner, it must be goal kick. It provides two information. First is few labels are wrongly added. Second is offensive events should be analyzed as type_name-2.




```
type_name-2
cross                4239
shot                 2360
pass                  537
shot_freekick         123
dribble                47
corner_crossed         25
corner_short           21
freekick_crossed       20
take_on                20
clearance              18
throw_in               17
keeper_save            11
tackle                  9
interception            6
freekick_short          4
goalkick                1
Name: Unnamed: 0, dtype: int64
```

Graph (1): Event: Interception

Clearance

Total Count : 56.790

Pre-Corner Count: 5.219

Rate : 9.2%

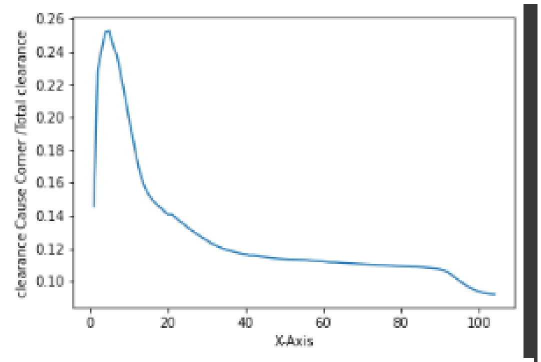
In first graph, number of corners that are caused by clearance has peak, but in correlation analysis, there is no correlation.

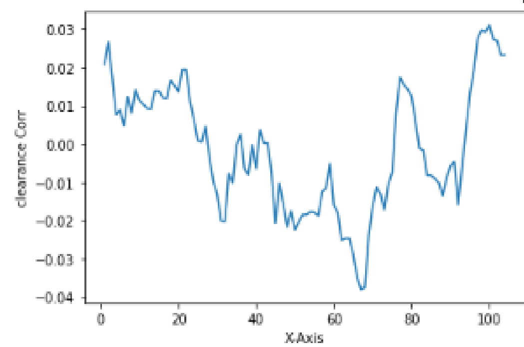
Correlations:

Clearance 22-31 (Corr: -0.066): It is valid, because it defines good defensive plays that obstruct opponents approach to keep.

Clearance 92-98 (Corr: 0.060): It is valid, because interval is close to the keep, and it is panic zone for defenders which means they need to play safe.

Lastly, events cause interceptions that causes corners are listed. Cross and pass take great part of table.






```
type_name-2
cross              2535
pass               1201
corner_crossed      281
freekick_crossed    265
shot                198
dribble             176
interception        153
take_on             113
throw_in             65
corner_short         51
tackle               45
keeper_save          35
clearance            34
shot_freekick        34
freekick_short       28
goalkick              5
Name: Unnamed: 0, dtype: int64
```

Graph (2): Event: Clearance

Keeper Save

Total Count : 12.531

Pre-Corner Count: 2.323

Rate : 18.53%

First graph has expected shape, but no information it provides.

Correlations:

Keeper Save 5-9 (Corr: -0.045): It may be valid, but correlation is too low. It will be analyzed at feature selection section.

1 <= keeper_save <= 9	-0.010980
2 <= keeper_save <= 9	-0.010430
3 <= keeper_save <= 9	-0.016367
4 <= keeper_save <= 9	-0.033614
5 <= keeper_save <= 9	-0.045196
6 <= keeper_save <= 9	-0.041025
7 <= keeper_save <= 9	-0.044319
8 <= keeper_save <= 9	-0.027987

Table (5): Related Correlation Values of keeper_save

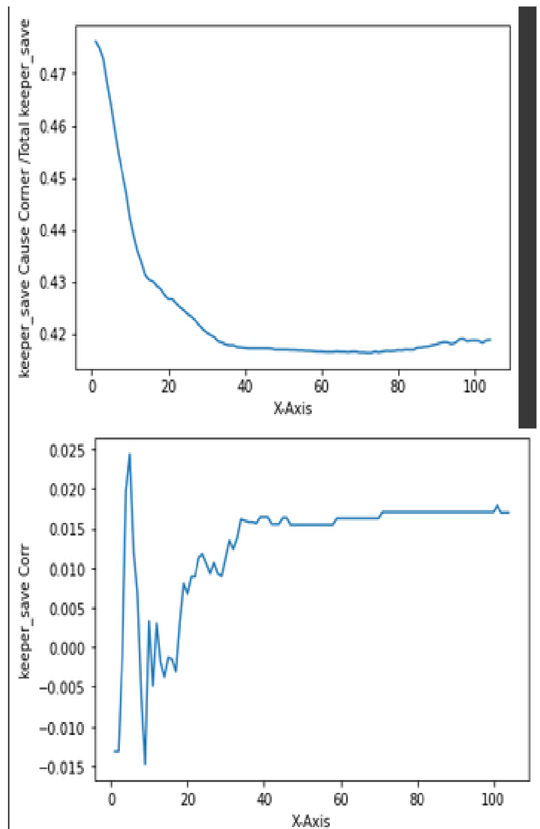
Keeper Savel 17-19 (Corr: 0.0545): It is not valid, because its interval is 1 mt and neighbor values are drop dramatically and its interval is too narrow.

Its only explanation can be done that it shows opponent team play counter attack, then keeper face with ball at line of penalty zone.

15 <= keeper_save <= 19	0.034013
16 <= keeper_save <= 19	0.037348
17 <= keeper_save <= 19	0.054564
18 <= keeper_save <= 19	0.030317

Table (6): Related Correlation Values of keeper_save

Lastly, events cause keeper save that causes corners are listed. Shot event dominate.




```
type_name-2
shot              2004
shot_freekick      155
cross               49
interception        21
dribble             20
shot_penalty        19
clearance           16
corner_crossed      13
freekick_crossed    12
pass                12
keeper_save          2
Name: Unnamed: 0, dtype: int64
```

Graph (3): Event: Keeper Save

Shot

Total Count : 43.071

Pre-Corner Count: 4.604 (pre-defensive events added)

Rate : 10.69%

First graph has expected shape, but no information it provides.

Correlations:

Shot 19-22 (Corr: 0.079): It is valid.

Cross

Total Count : 62.326

Pre-Corner Count: 7.252 (pre-defensive events added)

Rate : 11.64%

First graph has expected shape, but no information it provides.

Correlations:

Cross 6-14 (Corr: 0.072): It is valid. If cross is done below 6, probably probability of being goal kick is increasing. Also, it represents crossing close to zero line which are sometimes finish with interception and then corner.

Cross 40-51 (Corr: -0.0605): It is valid, because it represents early crosses, and they are generally finish with clearance.

Pass

In the first graph, it is obvious information that after few meter away from keep, passes don't cause corner.

Total Count : 1.646.227

Pre-Corner Count: 10.415 (pre-defensive events added)

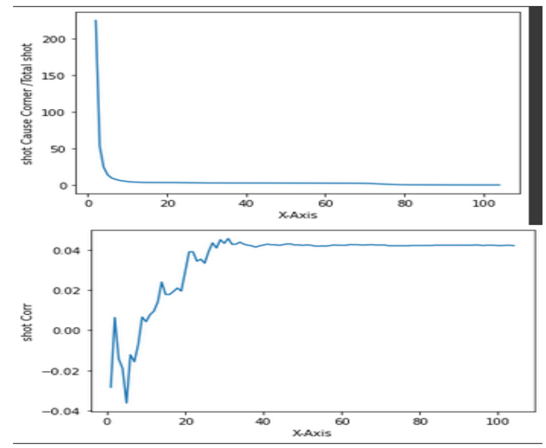
Rate : 0.63%

First graph has expected shape, but no information it provides.

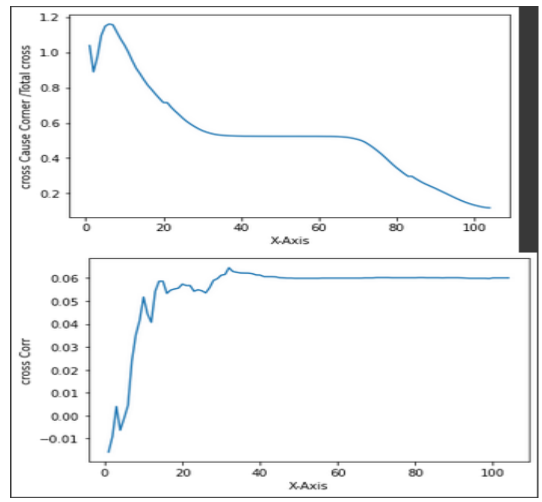
Correlations:

Pass 53-90 (Corr: -0.078): It is valid, because passes in the interval shows that teams cannot enter dangerous zone.

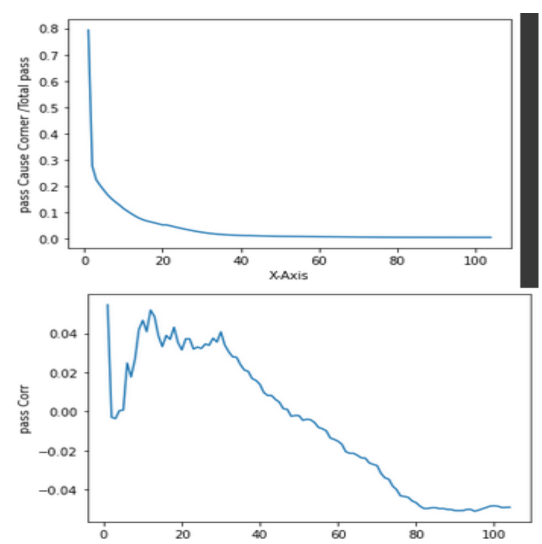
Pass 0-12 (Corr: 0.052): It is valid, because passes in the interval cause interception and then corner.



Graph (4): Event: Shot



Graph (5): Event: Cross



Graph (6): Event: Pass

Dribbling

Total Count : 194.477

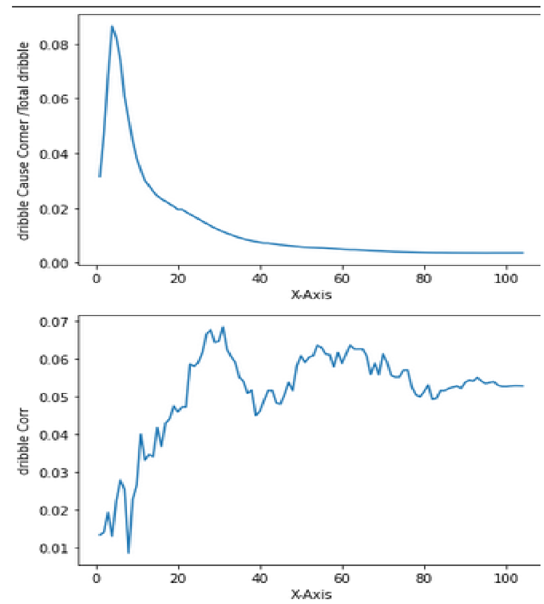
Pre-Corner Count: 679 (pre-defensive events added)

Rate : 0.34%

First graph tells us that if a player dribble to the zero line, it increases chance of corner.

Correlations:

Dribbling 8-28 (Corr: 0.071): Its validation will be analyzed at feature selection section.



Graph (7): Event: Dribbling

Possession (Time)

Correlations:

Pos_Time 34-94 (Corr: -0.082): It is valid, because it represent running down the clock.

Pos_Time 3-34 (Corr: 0.065): It is valid, because it means the ball is played in dangerous zone which is most frequently available for shooting, crossing etc.

Pos_Time 94-100 (Corr: 0.075): It is valid, because it is also dangerous zone for corner.

Posession (Event Count)

Correlations:

Pos_Count 95-100 (Corr: 0.072): It is valid, because it is possibly high potential corner zone.

Pos_Count 49-82 (Corr: -0.073): It is valid, because it represent running down the clock.

Pos_Count 4-30 (Corr: 0.071): It is valid, because it means the ball is played in dangerous zone which is most frequently available for shooting, crossing etc.

Posession – Feature Generation

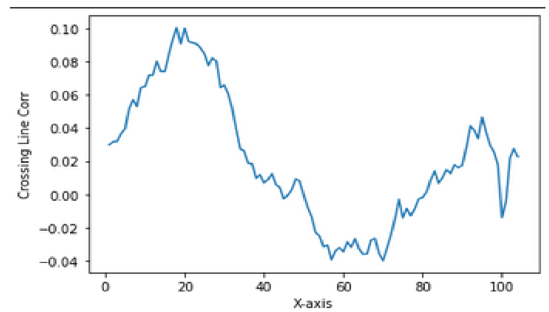
Pos_time in interval 3-34 and 94-100 and Pos_count in interval 4-30 and 95-100 provides positive correlations. Features are created by summation of them with time and count grouping.

Crossing Line

Crossing line is calculated as in and out.

Correlations:

Crossing Line 18 (Corr: 0.100): It is valid,because it means the ball is played in dangerous zone which is most frequently available for shooting, crossing etc. Nonetheless, it does not provide any information about dead zones.



Graph (8): Crossing Line

Goal Differences

Correlations:

Goals Corr : -0.042

Goals Difference Corr : -0.081

Goal difference makes sense, because if it increase, players drop tempo of game and wait for end.

Summary of EDA

During 1st EDA, events that cause corner are analyzed. These events' correlation with target corner is optimized by using correlation table which is varied with different x-axis intervals. The most correlated intervals are selected. The list is shown below,

	Features	X-axis Intervals	Correlations
Defensive Events	Interception	40 - 46	-0.077
		83 - 100	0.088
	Clearance	22 - 31	-0.066
		92 - 98	0.060
	Keeper Save	5 - 9	-0.045
		17 - 19	0.055
Offensive Events	Shot	19 - 22	0.079
	Pass	53 - 90	-0.078
		0 - 12	0.052
	Dribbling	8 - 28	0.071
Possession	Time	34 - 94	-0.082
		3 - 34	0.065
		94 - 100	0.076
		34-94 + 3-34	0.096
	Event Count	4 - 30	0.071
		49 – 82	-0.074
		95 - 100	0.072
		4-30 + 95-100	0.082
Crossing Line	In and Out	18	0.101
Goals	Goals Total	N/A	-0.042
	Goals Difference	N/A	-0.081
Corners	Realized Corner	N/A	0.015

Table (7): Summary of Created Features

4. Final Dataset

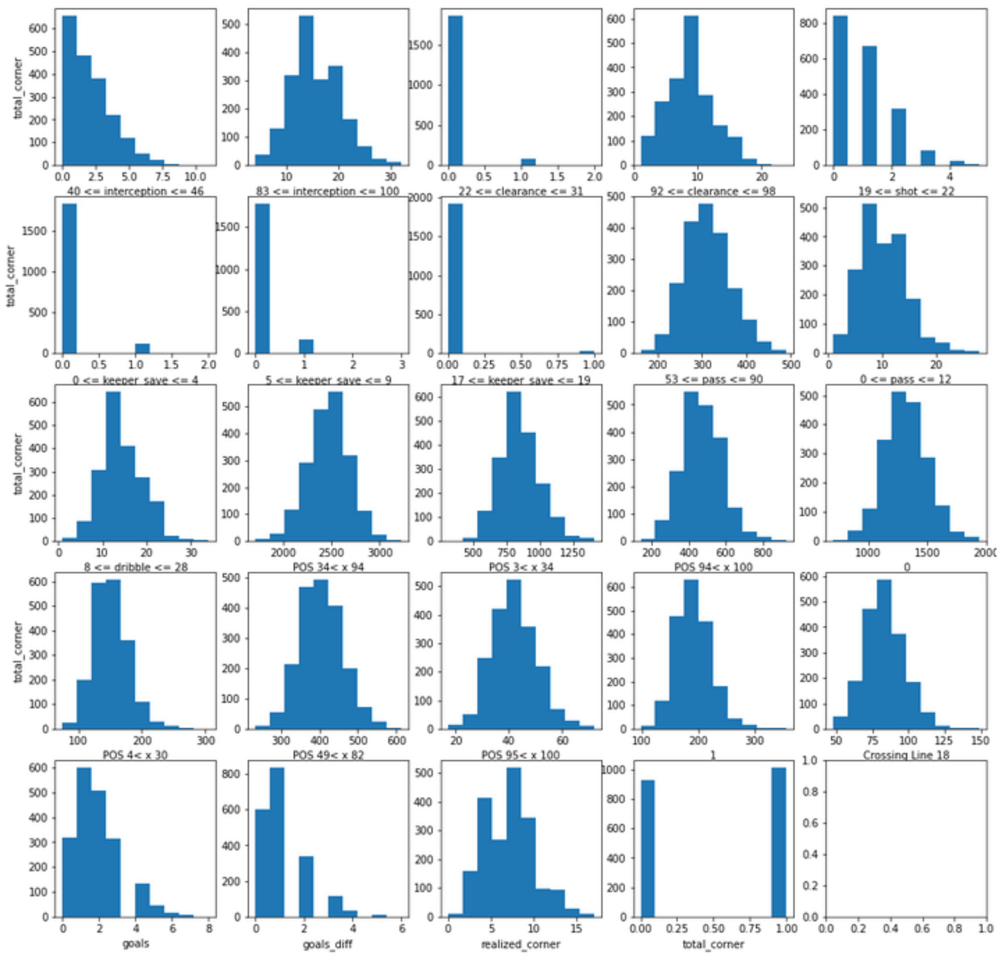
a. Dataset

	40 <= interception <= 46	83 <= interception <= 100	22 <= clearance <= 31	92 <= clearance <= 98	19 <= shot <= 22	0 <= keeper_save <= 4	5 <= keeper_save <= 9	17 <= keeper_save <= 19	53 <= pass <= 90	0 <= pass <= 12	8 <= dribble <= 28	POS 34< x 94	POS 3< x 34	POS 94< x 100	0	POS 4< x 30	POS 49< x 82	POS 95< x 100	1	Crossing Line 18	goals	goals_diff	realized_corner	total_corner
1694390	4.0	13	0.0	9	0.0	0.0	1.0	0.0	325	4	10	2355.770474	937.421359	518.414257	1455.835616	135	426	37	172	89	2	0	8.0	0.0
1694391	1.0	18	0.0	10	0.0	0.0	0.0	1.0	333	12	18	2509.762669	890.279594	373.541191	1263.820785	147	386	34	181	95	1	1	6.0	0.0
1694392	4.0	19	0.0	15	0.0	0.0	0.0	0.0	288	16	13	2346.311373	1061.274311	424.708643	1485.982954	165	370	51	216	91	2	0	10.0	0.0
1694393	2.0	14	0.0	6	0.0	0.0	0.0	0.0	352	3	14	2228.981802	926.156163	608.456832	1534.612995	128	443	42	170	65	0	0	8.0	0.0
1694394	2.0	15	0.0	17	1.0	0.0	0.0	0.0	301	10	19	2387.098325	928.349460	305.361191	1233.710651	162	370	38	200	91	0	0	15.0	0.0

Table (8): First 5 rows of final dataset

The dataframe which is shown above is created. It has 1941 rows and 16 columns without any missing values.

b. Histogram

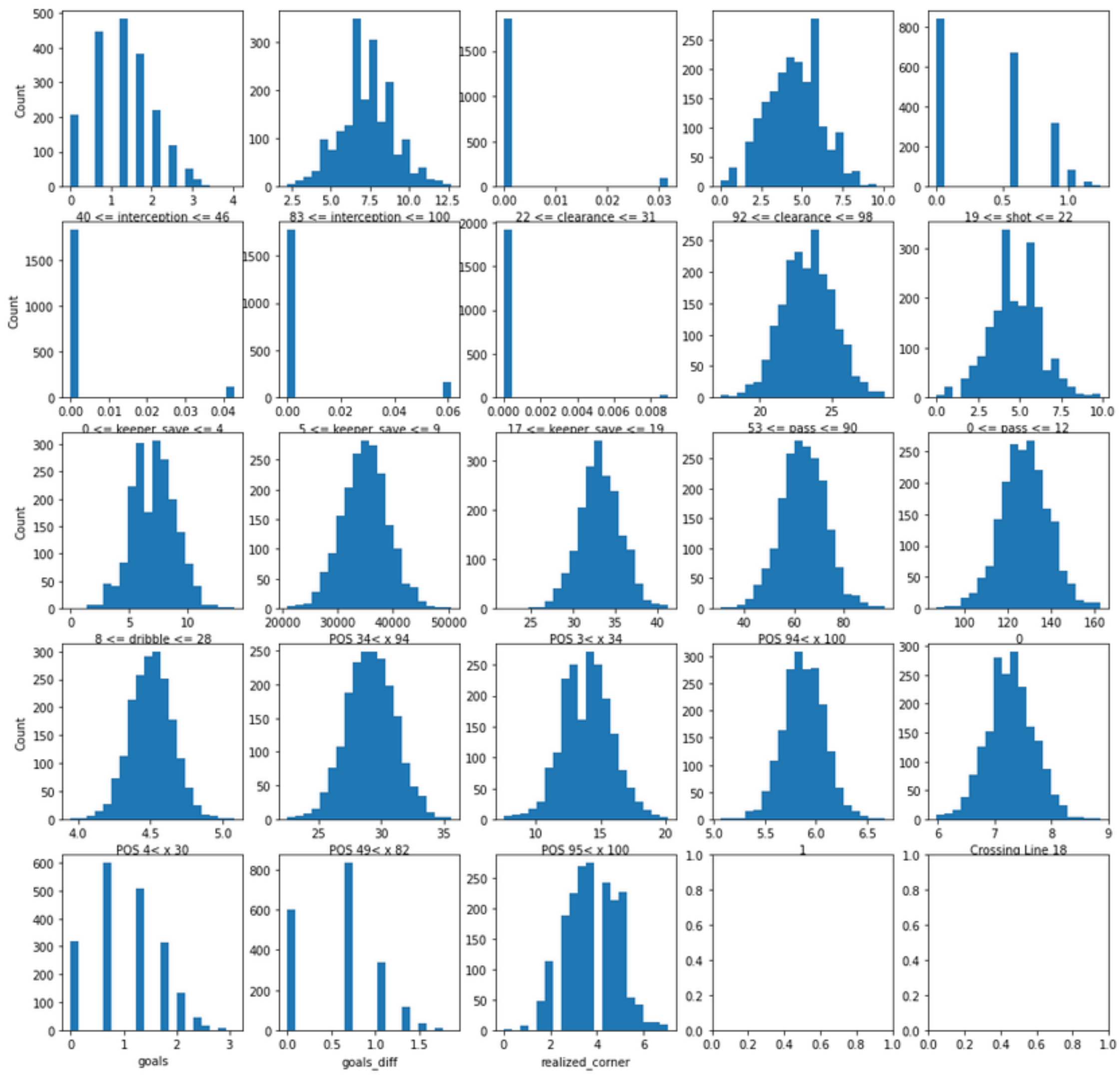


Graph (9): Histogram plot of final dataset

5. **Transformation**

Box-Cox transformation is applied. The transformation is applied to subset of features, because of high number of zero values of some features. Transforming them cause bias.

After the transformation is done, histogram becomes as below,



Graph (10): Histogram Plot of Transformed Final Dataset

To test normalization of features, Shapiro – Wilk Test is applied.

Feature Name	Stats	P-Value
40 <= interception <= 46	0.950	3.648 e-25
83 <= interception <= 100	0.9941	6.776 e-07
22 <= clearance <= 31	0.20844	0.0
92 <= clearance <= 98	0.99168	4.594 e-09
19 <= shot <= 22	0.799948	8.688 e-44
0 <= keeper_save <= 4	0.2522553	0.0
5 <= keeper_save <= 9	0.313498	0.0
17 <= keeper_save <= 19	0.086741	0.0
53 <= pass <= 9	0.999319	0.731
0 <= pass <= 12	0.9932613	9.339 e-08
8 <= dribble <= 28	0.99511	5.565 e-06
POS 34< x 94	0.999287	0.691
POS 3< x 34	0.9980944	0.0226
POS 94< x 100	0.9985110	0.084
0	0.99861	0.118
POS 4< x 30	0.998702	0.152
POS 49< x 82	0.999515	0.929
POS 95< x 100	0.99763	0.005
1	0.998378	0.0557
Crossing Line 18	0.9985	0.0901
goals	0.925354	5.0585 e-30
goals_diff	0.853143	3.514 e-39
realized_corner	0.98737	4.968 e-12

Table (9): Shapiro – Wilk Test Results

Reds can be considered as categorical variables, because they vary between few values. That's the reason why these cannot be normalized. Other features are normalized as expected.

Secondly, Standard Scaler is applied to dataframe.

6. Handling Missing Values

All values in each row carries realized in-game data, so outliers of data do not consist of noise or misinformation. Thus, outliers aren't exposed to any processing method.

7. **Feature Selection**

To choose the most effective features, Sequential Feature Selector function is used. It is applied before each model to specify their features.

8. **Model Selection**

To identify minimum required goal, calculations below are done,

	Count	**Probability**	**Betting Odds (with 7% Commision)**
y == 1 (over 2.5)	1012	52.14%	1.80 (55.64%)
y == 0 (under 2.5)	929	47.86%	1.95 (51.36%)

Table (10): Main Goal Metrics – Odds for Each Outcome

If a model predicts all 1, it gets 52.14% accuracy score and it gives 3.5% commission. Therefore, our main target is to predict number of corner with at least 55.64% accuracy.

To select the highest performed models, at first, LazyClassifier is applied to dataset. LazyClassifier is a method to apply 27 base models to select most fitted ones.

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
NearestCentroid	0.57	0.57	0.57	0.57	0.02
GaussianNB	0.57	0.57	0.57	0.57	0.01
SVC	0.57	0.57	0.57	0.57	0.28
ExtraTreesClassifier	0.57	0.57	0.57	0.57	0.30
BernoulliNB	0.56	0.56	0.56	0.56	0.02
LinearDiscriminantAnalysis	0.56	0.56	0.56	0.56	0.04
LogisticRegression	0.56	0.56	0.56	0.56	0.05
CalibratedClassifierCV	0.56	0.55	0.55	0.55	0.61
RidgeClassifierCV	0.55	0.55	0.55	0.55	0.05
RidgeClassifier	0.55	0.55	0.55	0.55	0.03
LinearSVC	0.55	0.55	0.55	0.55	0.23
RandomForestClassifier	0.55	0.55	0.55	0.55	0.51
NuSVC	0.55	0.55	0.55	0.55	0.30
DecisionTreeClassifier	0.54	0.54	0.54	0.54	0.05
KNeighborsClassifier	0.54	0.54	0.54	0.54	0.09
QuadraticDiscriminantAnalysis	0.53	0.53	0.53	0.53	0.05
PassiveAggressiveClassifier	0.52	0.53	0.53	0.51	0.02
BaggingClassifier	0.52	0.53	0.53	0.52	0.17
XGBClassifier	0.53	0.52	0.52	0.52	0.16
ExtraTreeClassifier	0.52	0.52	0.52	0.52	0.02

Table (11): Result of Lazy Classifier

During model selection, not only accuracy score, but also different approach to diversify model types for using them in ensembling is taken into account. Lastly, having predict_proba function is considered.
K-Neighbor Classifier, Gaussian Naïve Bayes, Support Vector Classifier, Extra Trees Classifier, Logistic Regression and Stochastic Gradient Decent Classifier are selected.

a. K-Neighbors Classifier

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
          '22 <= clearance <= 31',      '92 <= clearance <= 98',
              '19 <= shot <= 22',      '0 <= keeper_save <= 4',
          '5 <= keeper_save <= 9',   '17 <= keeper_save <= 19',
              '53 <= pass <= 90',            '0 <= pass <= 12',
            '8 <= dribble <= 28',              'POS 34< x 94',
                   'POS 3< x 34',             'POS 94< x 100',
                              0,                'POS 4< x 30',
                  'POS 49< x 82',             'POS 95< x 100',
                              1,           'Crossing Line 18',
                         'goals',                'goals_diff',
             'realized_corner'],
      dtype='object')
```

```
accuracy_score on test dataset :  0.5318627450980392
[[ 95  91]
 [100 122]]
{'leaf_size': 1, 'n_neighbors': 3, 'p': 2}
              precision    recall  f1-score   support

         0.0       0.49      0.51      0.50       186
         1.0       0.57      0.55      0.56       222

    accuracy                           0.53       408
   macro avg       0.53      0.53      0.53       408
weighted avg       0.53      0.53      0.53       408
```

Table (12): Selected Features and Model Results

b. Gaussian Naïve Bayes

```
Fitting 15 folds for each of 100 candidates, totalling 1500 fits
accuracy_score on test dataset :  0.5523156089193825
[[143 137]
 [124 179]]
{'var_smoothing': 1.2328467394420635e-09}
              precision    recall  f1-score   support

         0.0       0.54      0.51      0.52       280
         1.0       0.57      0.59      0.58       303

    accuracy                           0.55       583
   macro avg       0.55      0.55      0.55       583
weighted avg       0.55      0.55      0.55       583
```

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
          '22 <= clearance <= 31',      '92 <= clearance <= 98',
              '19 <= shot <= 22',      '0 <= keeper_save <= 4',
          '5 <= keeper_save <= 9',   '17 <= keeper_save <= 19',
              '53 <= pass <= 90',            '0 <= pass <= 12',
            '8 <= dribble <= 28',              'POS 34< x 94',
                   'POS 3< x 34',             'POS 94< x 100',
                              0,                'POS 4< x 30',
                  'POS 49< x 82',             'POS 95< x 100',
                              1,           'Crossing Line 18',
                         'goals',                'goals_diff',
             'realized_corner'],
      dtype='object')
```

Table (13): Selected Features and Model Results

c. Support Vector Classifier

```
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
SVC(C=10, gamma=0.1)
[[  2 278]
 [  1 302]]
              precision    recall  f1-score   support

         0.0       0.67      0.01      0.01       280
         1.0       0.52      1.00      0.68       303

    accuracy                           0.52       583
   macro avg       0.59      0.50      0.35       583
weighted avg       0.59      0.52      0.36       583
```

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
          '22 <= clearance <= 31',      '92 <= clearance <= 98',
              '19 <= shot <= 22',      '0 <= keeper_save <= 4',
          '5 <= keeper_save <= 9',   '17 <= keeper_save <= 19',
              '53 <= pass <= 90',            '0 <= pass <= 12',
            '8 <= dribble <= 28',              'POS 34< x 94',
                   'POS 3< x 34',             'POS 94< x 100',
                              0,                'POS 4< x 30',
                  'POS 49< x 82',             'POS 95< x 100',
                              1,           'Crossing Line 18',
                         'goals',                'goals_diff',
             'realized_corner'],
      dtype='object')
```

Table (14): Selected Features and Model Results

d. Extra Trees Classifier

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
accuracy_score on test dataset :  0.5728987993138936
[[132 148]
 [100 203]]
{'n_estimators': 1200, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_features': 'log2', 'max_depth': 20, 'criterion': 'entropy', 'bootstrap': False}
              precision    recall  f1-score   support

         0.0       0.57      0.47      0.52       280
         1.0       0.58      0.67      0.62       303

    accuracy                           0.57       583
   macro avg       0.57      0.57      0.57       583
weighted avg       0.57      0.57      0.57       583
```

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
              '22 <= clearance <= 31',      '92 <= clearance <= 98',
                 '19 <= shot <= 22',      '0 <= keeper_save <= 4',
              '5 <= keeper_save <= 9',   '17 <= keeper_save <= 19',
                 '53 <= pass <= 90',            '0 <= pass <= 12',
                '8 <= dribble <= 28',            'POS 34< x 94',
                     'POS 3< x 34',             'POS 94< x 100',
                                0,               'POS 4< x 30',
                    'POS 49< x 82',             'POS 95< x 100',
                                1,          'Crossing Line 18',
                          'goals',                'goals_diff',
                  'realized_corner'],
      dtype='object')
```

Table (15): Selected Features and Model Results

e. Logistic Regression

```
accuracy_score on test dataset :  0.5574614065180102
[[136 144]
 [114 189]]
{'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
              precision    recall  f1-score   support

         0.0       0.54      0.49      0.51       280
         1.0       0.57      0.62      0.59       303

    accuracy                           0.56       583
   macro avg       0.56      0.55      0.55       583
weighted avg       0.56      0.56      0.56       583
```

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
              '22 <= clearance <= 31',      '92 <= clearance <= 98',
                 '19 <= shot <= 22',      '0 <= keeper_save <= 4',
              '5 <= keeper_save <= 9',   '17 <= keeper_save <= 19',
                 '53 <= pass <= 90',            '0 <= pass <= 12',
                '8 <= dribble <= 28',            'POS 34< x 94',
                     'POS 3< x 34',             'POS 94< x 100',
                                0,               'POS 4< x 30',
                    'POS 49< x 82',             'POS 95< x 100',
                                1,          'Crossing Line 18',
                          'goals',                'goals_diff',
                  'realized_corner'],
      dtype='object')
```

Table (16): Selected Features and Model Results

f. Ensemble Models

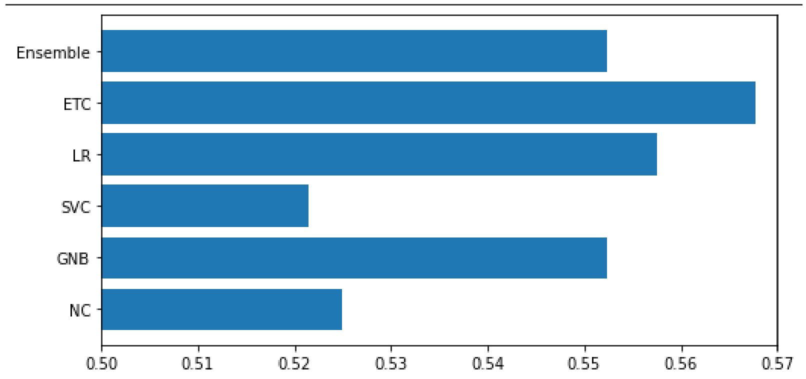
As ensemble model, VotingClassifier is used with soft voting.

```
accuracy_score on test dataset :  0.5523156089193825
              precision    recall  f1-score   support

         0.0       0.54      0.48      0.51       280
         1.0       0.56      0.62      0.59       303

    accuracy                           0.55       583
   macro avg       0.55      0.55      0.55       583
weighted avg       0.55      0.55      0.55       583

[[134 146]
 [115 188]]
```

```
VotingClassifier(estimators=[('KNN',
                              KNeighborsClassifier(leaf_size=1, n_neighbors=1)),
                             ('GNB',
                              GaussianNB(var_smoothing=1.2328467394420635e-09)),
                             ('ETC',
                              ExtraTreesClassifier(criterion='entropy',
                                                   max_depth=20,
                                                   min_samples_leaf=2,
                                                   min_samples_split=10,
                                                   n_estimators=600)),
                             ('LR',
                              LogisticRegression(C=100, solver='newton-cg')),
                             ('SVC', SVC(C=10, gamma=0.1, probability=True))],
                 voting='soft')
```

Table (17): Selected Features and Model Results

Accuracy Result Comparison,



Graph (11): Comparison of Models

g. Recall Adjustment

The core aim of the project is making accuracy highest. If passing some games without prediction is considered, predict_proba function of models will be a good tool to sort wrong predictions out.

Main strategy is to select higher probability that are taken from predict_proba for increasing recall.

In this part, Voting Classifier is taken into account.

```
For 0's
Best Recall is: 1.0
Best prob value is: 69
[[18  0]
 [13  0]]
              precision    recall  f1-score   support

         0.0       0.58      1.00      0.73        18
         1.0       0.00      0.00      0.00        13

    accuracy                           0.58        31
   macro avg       0.29      0.50      0.37        31
weighted avg       0.34      0.58      0.43        31

For 1's
Best Recall is: 1.0
Best prob value is: 67
[[ 1 25]
 [ 0 43]]
              precision    recall  f1-score   support

         0.0       1.00      0.04      0.07        26
         1.0       0.63      1.00      0.77        43

    accuracy                           0.64        69
   macro avg       0.82      0.52      0.42        69
weighted avg       0.77      0.64      0.51        69

[[20 29]
 [13 58]]
              precision    recall  f1-score   support

         0.0       0.61      0.41      0.49        49
         1.0       0.67      0.82      0.73        71

    accuracy                           0.65       120
   macro avg       0.64      0.61      0.61       120
weighted avg       0.64      0.65      0.63       120
```

Table (18): Filtered Result for Each Outcome and Total Filtered Result

Best probability values for each outcome is found as 69% for 1 and 67% for 0. If the model predicts a game's probability between 33% and 67%, it will ignore and will not bet to it. If it is higher, then it bets. Its accuracy is increased to 0.65. It ignores 463 games and take into account 120 games.

9. Conclusion

Main aim of the project is defined as predicting soccer games' corner count after a specific moment. To define evaluation metric, bookmakers' ratio calculation method is analyzed. Main goal is determined that getting better accuracy performance than 55.5% (52% + 3.5%).

Data engineering to select the most informative feature is done. During the data engineering phase, players' event types are analyzed. In order to decrease actions that have low probability to make corner, x-axis intervals are researched. Also, crossing line count, goal count, goal differences, realized corner count and some feature creations are considered. Final dataset that includes features listed below,

```
Index([ '40 <= interception <= 46', '83 <= interception <= 100',
        '22 <= clearance <= 31',       '92 <= clearance <= 98',
             '19 <= shot <= 22',       '0 <= keeper_save <= 4',
        '5 <= keeper_save <= 9',     '17 <= keeper_save <= 19',
             '53 <= pass <= 90',             '0 <= pass <= 12',
          '8 <= dribble <= 28',               'POS 34< x 94',
                 'POS 3< x 34',               'POS 94< x 100',
                             0,                 'POS 4< x 30',
                'POS 49< x 82',               'POS 95< x 100',
                             1,            'Crossing Line 18',
                       'goals',                 'goals_diff',
             'realized_corner',               'total_corner'],
      dtype='object')
```

To normalize features, Box-Cox transformation is applied. Due to behave some features ordinal, these features cannot be normalized. To use distance based models, standardization is done by using Standard Scaler. Target variable is converted from numerical to binary, because of imitating data to betting case.

In model selection phase, Lazy Classifier is used to determine the most fitted models to the dataset. K-Neighbors Classifier, Extra Trees Classifier, Logistic Regression, Support Vector Classifier and Gaussian Naïve Bayes are selected. After hyper parameter tuning, Voting Classifier is used to ensemble these models. The project goal allows us to leave some of games without predict. Therefore, just assured games are predicted. Assurance is determined by the light of predict_proba which is given by the ensemble model.
Finally, case of betting predicted games are simulated. By using confusion matrix, summary table is created below,

	Predicted Negative (1.95)	**Predictive Positive (1.80)**	**Bet for Each Match**	**Total Sum**
Actual Negative	20 (+190 TL)	29 (-290 TL)	10	-100 TL
Actual Positive	13 (-130 TL)	58 (+464 TL)	10	+334 TL
Total	+60 TL	+174 TL	1200 TL	+234 TL

The model should be tested newly generated data to be sure it is working. Things that should be done to develop model are listed below,

1. EDA of y-axis: During EDA in the project, only x-axis is analyzed. Y-axis is hiding a treasure in it.
2. Case studies: Soccer games should be watched and scenarios that causes corner should be represented by using data.
3. Best minutes check: In the project minute: 67.5 is used to divide dataset. Different time intervals should be tried to get better performance.