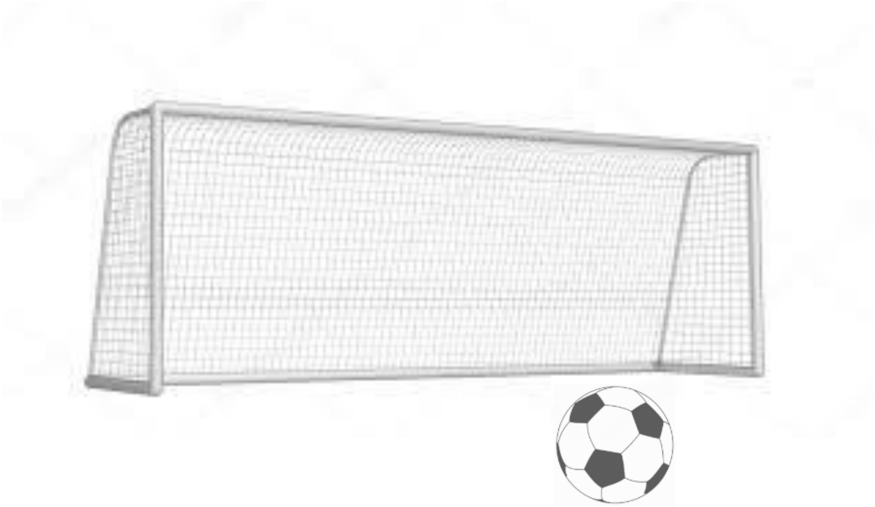
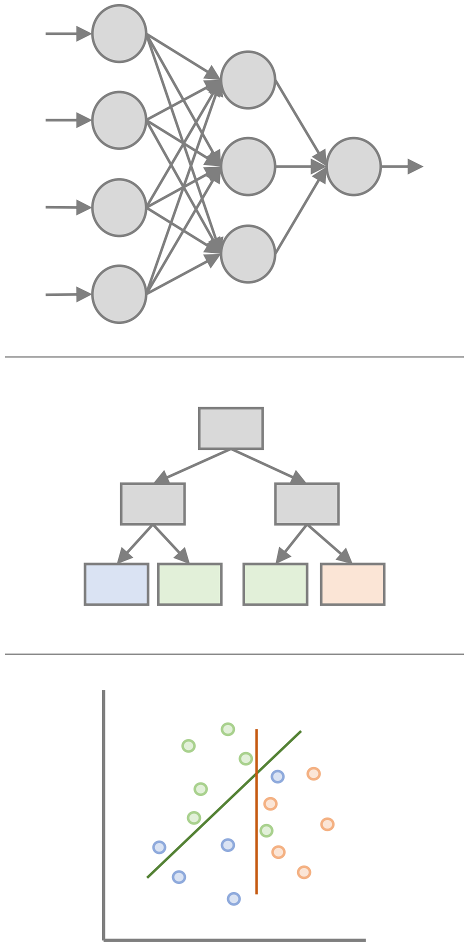




Predicting the 1. Bundesliga

Advanced Data Science with IBM - capstone project, Coursera

By Florian Parche

Overview

Idea:

- Predict the outcome[1] of 1. Bundesliga[2] games

How?

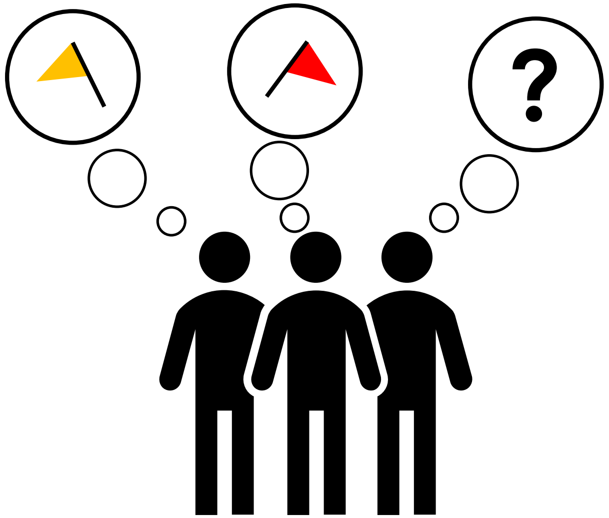
- Machine Learning
  - Neural Network
  - Decision Tree
  - Suppor Vector Machine

Goal:

- Outperform ‘naive’ picks
  - The favorite
  - The home team

Use case:

- Beat your friends & colleagues



1) Defined as: Home team win – Draw – Away team win

2) The 1. Bundesliga is the highest German soccer league

Result

Accuracy:

Neural Network:	53.52%
Favorite:	53.17%
Home Team:	49.47%

0.35%

The model barely and insignificantly outperforms ‘naive’ approaches

Data

Dataset[1]:

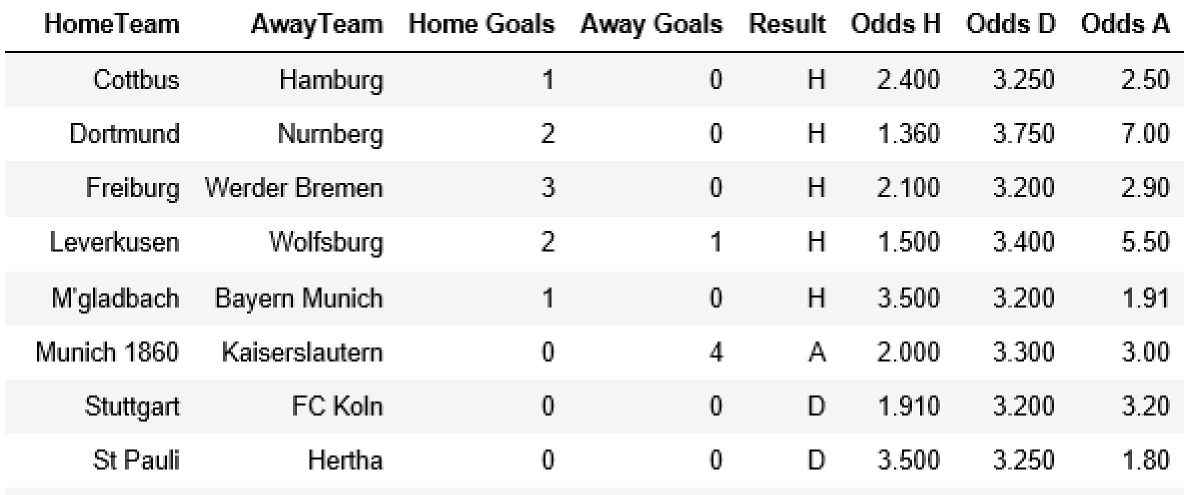
- 3,780 Games
- 28 Teams[2]
- Aug 2001 – May 2018

Inputs:

- Outcome Odds[3]
- Feature Engineering
    - Past goals scored
    - Past goals scored against
    - Past points per game
    - Ranks of the above (among all teams)

Exclusions:

- Games with either team not in the league in the previous season

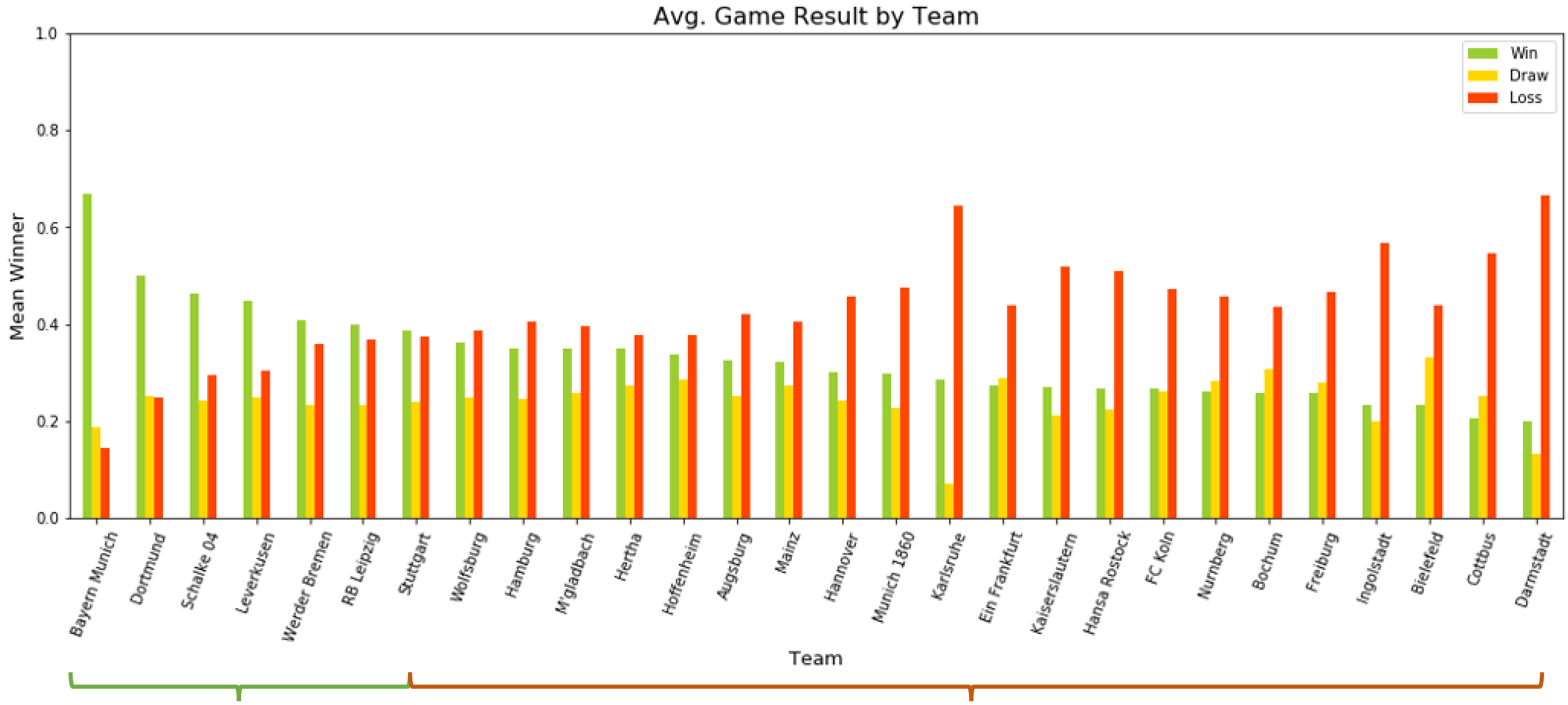


HomeTeam	AwayTeam	Home Goals	Away Goals	Result	Odds H	Odds D	Odds A
Cottbus	Hamburg	1	0	H	2.400	3.250	2.50
Dortmund	Nurnberg	2	0	H	1.360	3.750	7.00
Freiburg	Werder Bremen	3	0	H	2.100	3.200	2.90
Leverkusen	Wolfsburg	2	1	H	1.500	3.400	5.50
M'gladbach	Bayern Munich	1	0	H	3.500	3.200	1.91
Munich 1860	Kaiserslautern	0	4	A	2.000	3.300	3.00
Stuttgart	FC Koln	0	0	D	1.910	3.200	3.20
St Pauli	Hertha	0	0	D	3.500	3.250	1.80

Cleaned data prior to feature engineering

1) Data source: www.football-data.co.uk/germanym
2) Note that there are 18 Teams in the league in any given season
3) Odds are the median from three providers that are available in every season in the dataset

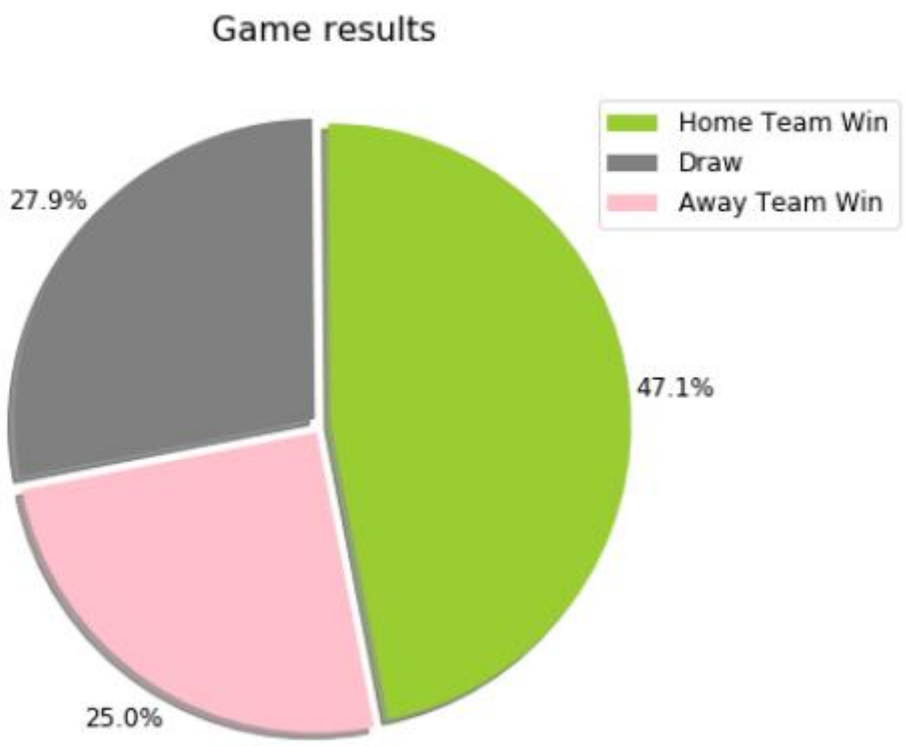
Historical Winners

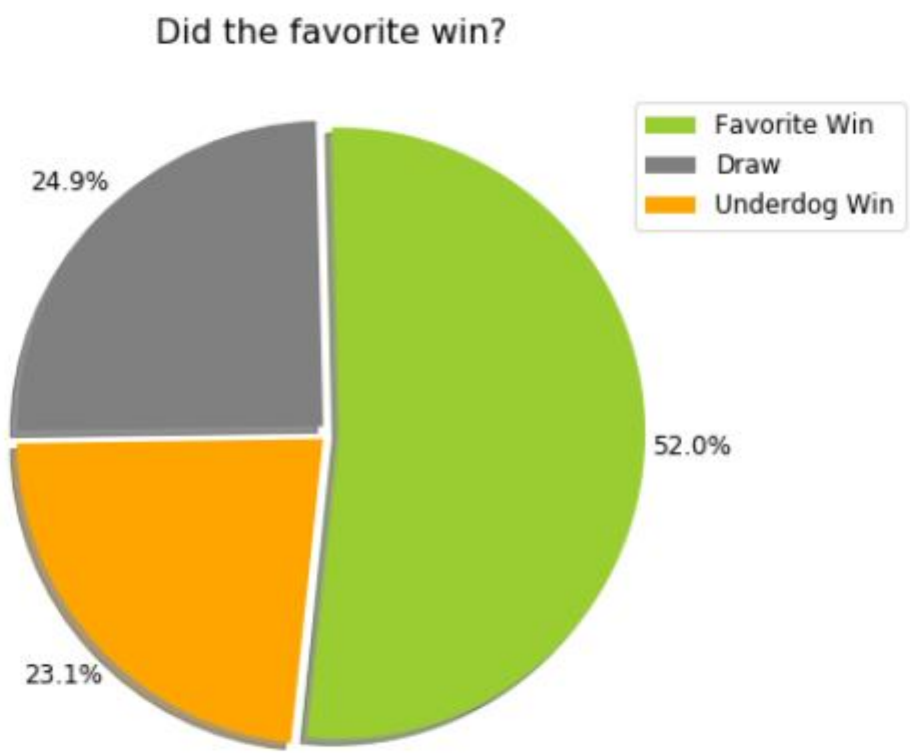


7 Teams: W > L

21 Teams: W < L

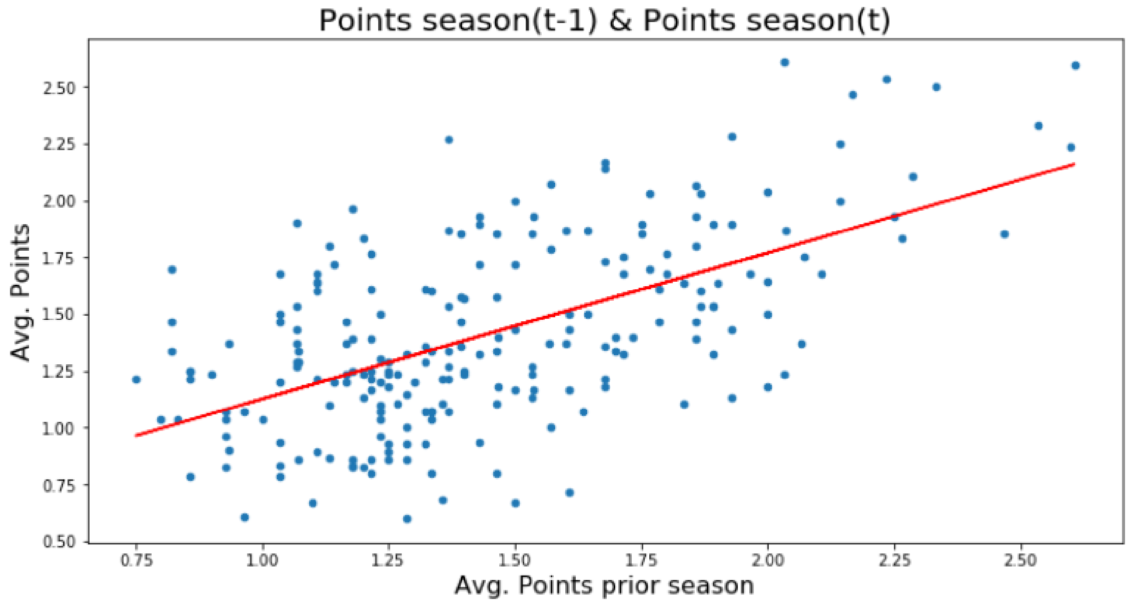
Historical Tendencies

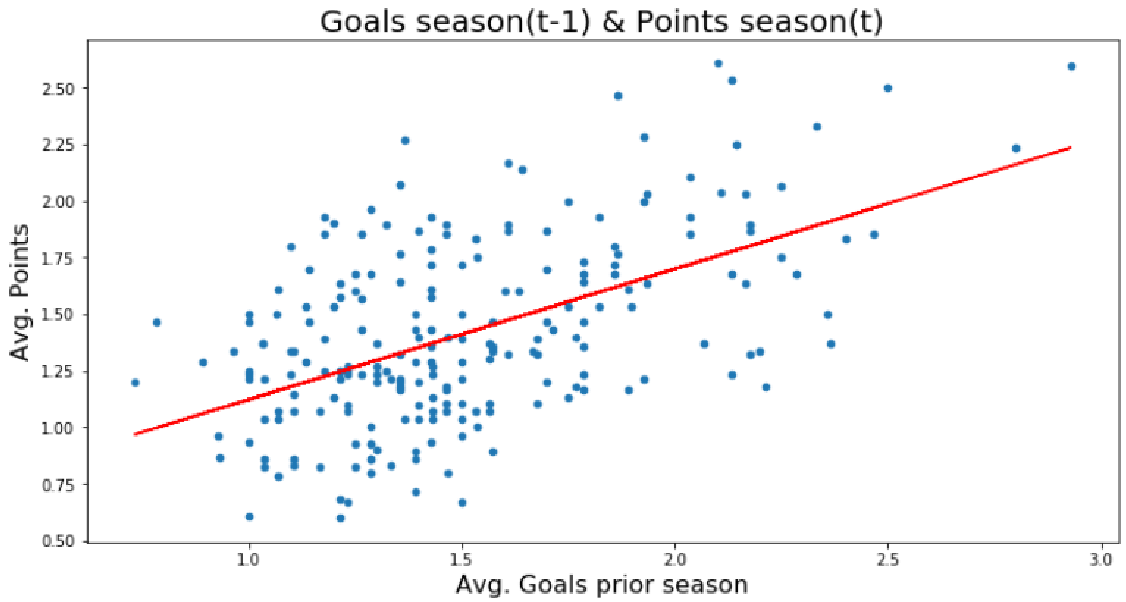




- The home team tends to win
- The favorite tends to win

Predictability





- Eyeball-Econometrics: There is some predictability

Feature Engineering

General:

- Prev. Season:
  - Total goals scored (H & A)
  - Total goals scored against (H & A)
  - Points at season end

Home Team:

- Goals scored when at home
- Goals scored against when at home
- H, D, A odds

Away Team:

- Goals scored when away
- Goals scored against when away
- H, D, A odds

▶ **47 Features** in total

- Prev. season's statistics
- Avg. statistics prev. **n** games
- Avg. statistics prev. **m** games ranked among all teams

Hyperparameters

General

- Weight of prev. season‘s statistics
- No. of prev. games for averages
- No. of prev. games for ranks
- Scaler[1]

Neural Network

- No. nodes in dense layer[2]
- Dropout rate[2]
- Activation function[3]
- Optimizer[4]
- Epochs & Batchsize

Decision Tree

- Max tree depth

SVM

- Kernel[5]
- If Polynomial - degree

[1] MinMax, StandardScaler or [-1, 1]

[2] The layers of the NN include 4 dense layers and 2 dropout layers

[3] Relu, leaky Relu or Tanh

[4] Adam, Adadelta, Adagrad or SGD

[5] Linear, Polynomial or Gaussian

Model results

Datasplit:

- 75% Training – 10% Validation – 15% Testing

Model	Accuracy	
Neural Network	**53.52%**	
Decision Tree	50.53%	
SVM	52.46%	
Democracy[1]	55.52%	
Unanimous[2]	**60.34%**	⚽
Favorite	**53.17%**	
Home Team	49.47%	🟥

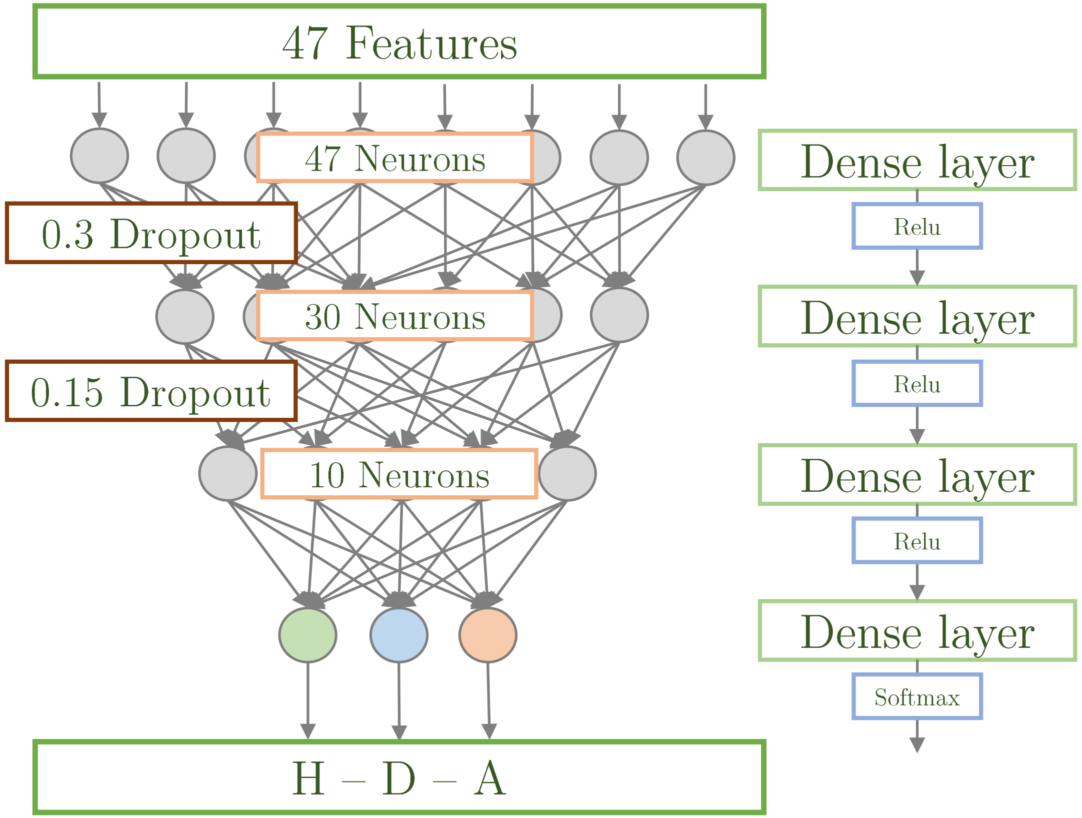
- ‘Unanimous’ is significantly more accurate (5% confidence level of better), except vs ‘Democracy’
- ‘Democracy’ is significantly more accurate than ‘Tree’ (10% level) and ‘Home Team’ (5%)
- ‘Neural Network’ is not significantly more accurate than any other approach

1) NN, Tree and SVM ’vote‘ which result to predict
2) NN, Tree and SVM predict the same. This approach picked 353 out of 567 (~62%) games in the test set.

Model details – Neural Network

Hyperparameters:

Weight of prev. Season:	100%
No. prev. Games for Avgs.:	15
No. prev. Games for Ranks:	15
Scaler:	StandardScaler
Optimizer:	Adam
Epochs:	40
Batchsize:	32



47 Features

47 Neurons

0.3 Dropout

30 Neurons

0.15 Dropout

10 Neurons

H – D – A

Dense layer

Relu

Dense layer

Relu

Dense layer

Relu

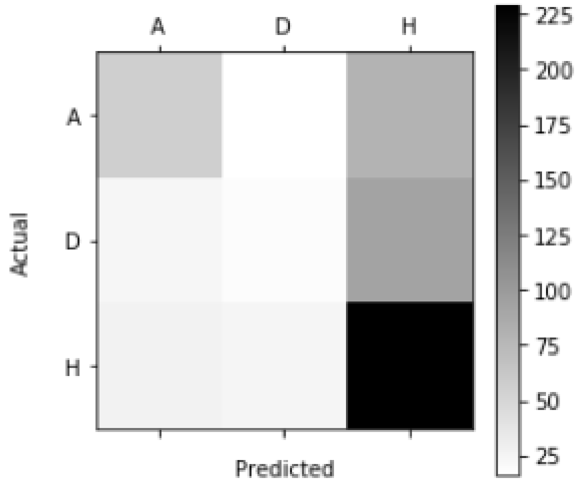
Dense layer

Softmax

Evaluation – Neural Network

- Test set: 567 Games – Accuracy: 53.52%

Confusion Matrix



Predicted	A	D	H
Actual			
A	56	16	80
D	24	19	92
H	27	25	229

NN predicts Win[1]

Team	NN for	NN for correct	%
Kaiserslautern	7	6	85.71
Bochum	7	5	71.43
Hamburg	25	17	68.00
Bayern Munich	74	49	66.22
Hertha	19	12	63.16

•
•
•

Team	NN for	NN for correct	%
Hannover	20	7	35.00
Werder Bremen	24	7	29.17
Freiburg	7	2	28.57
Mainz	14	4	28.57
Ein Frankfurt	13	2	15.38

NN predicts not-Win[1]

Team	NN against	NN against correct	%
Hansa Rostock	13	10	76.92
Freiburg	25	17	68.00
Cottbus	8	5	62.50
M'gladbach	35	21	60.00
Hannover	45	27	60.00

•
•
•

Team	NN against	NN against correct	%
Werder Bremen	35	15	42.86
Kaiserslautern	17	7	41.18
Stuttgart	27	11	40.74
Leverkusen	31	12	38.71
Hamburg	39	12	30.77

1) Filtered for 5 or more win / not-win predictions

Strategy evaluation

Strategy:

- Combine model prediction & teams that were accurately predicted in the training set
- Most accurate **n** teams picked to win, **m** teams picked to not-win
- No. Teams included for which
  $$\# \text{ Picks} * (\text{Accuracy} ** 2)$$
  is maximized in the validation set

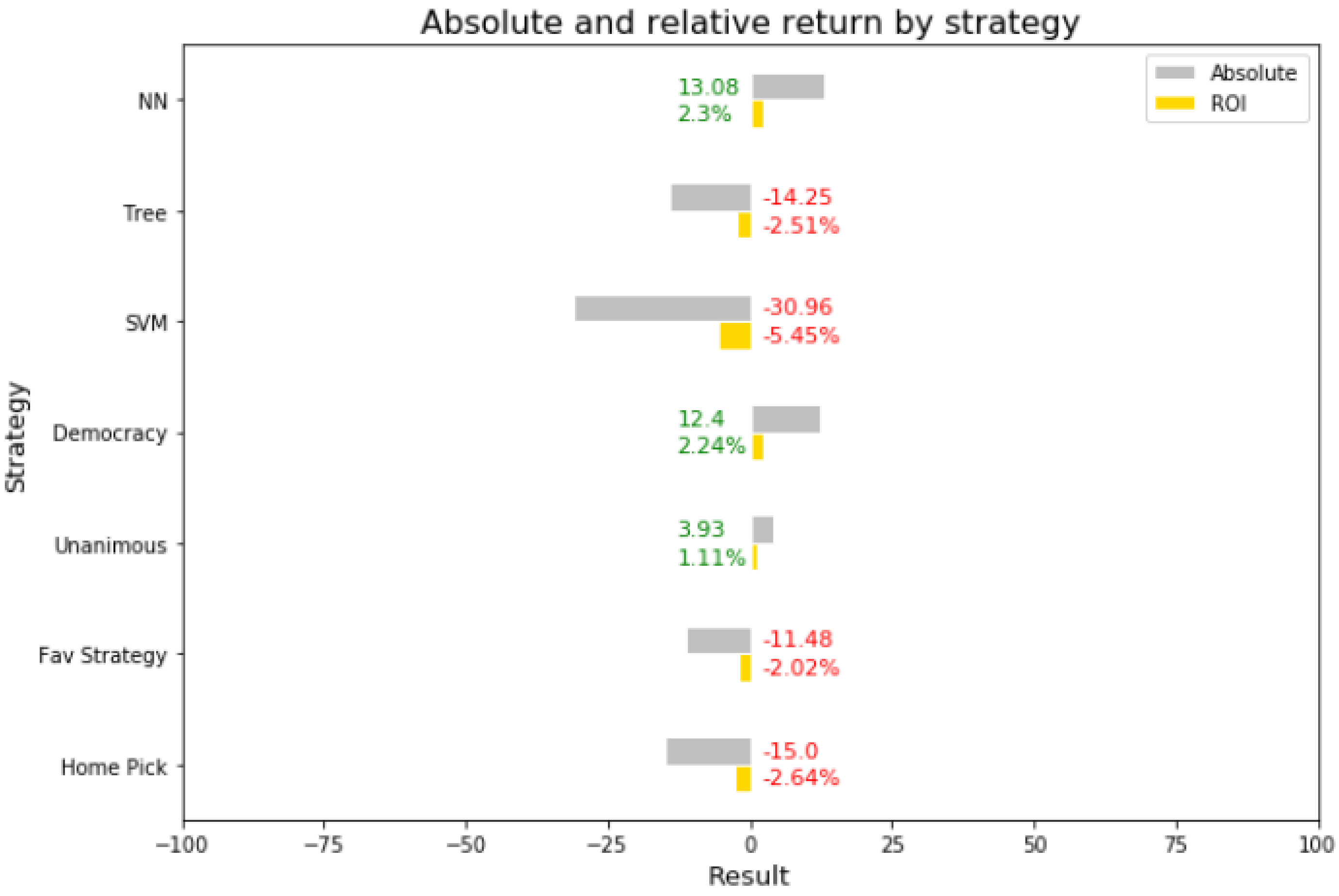
Model	Accuracy	Strategy[1]	
Neural Network	53.5%	55.0% (340)	⬆
Decision Tree	50.5%	52.2% (389)	⬆
SVM	52.5%	51.6% (401)	⬇
Democracy	55.5%	54.5% (268)	⬇
Unanimous	60.3%	61.3% (181)	⬆
Favorite	53.2%	53.9% (408)	⬆
Home Team	49.5%	47.5% (379)	⬇

[1] Number in parentheses is the number of picks. The maximum possible is 567 (the test set size)

Returns

If you were to bet 1 unit on every model prediction in the test set[1]

Model	Accuracy
Neural Network	53.52%
Decision Tree	50.53%
SVM	52.46%
Democracy	55.52%
Unanimous	60.34%
Favorite	53.17%
Home Team	49.47%



1) Note that the models are neither designed nor intended to take advantage of discrepancies in the estimated and for betting available odds

2018/19

Accuracy in the current season – does it work in practice?

Gameday

Model / Method

Accuracy in %	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
NN	57	43	71	43	57	38	43	43	29	29	29	57	43	57	71	43	57	71	48.8
Tree	43	43	43	29	43	38	43	43	43	14	43	57	43	43	43	43	86	71	44.9
SVM	43	14	71	29	57	38	43	29	29	43	57	57	43	29	71	29	71	57	44.9
Democracy	50	33	67	33	57	38	43	33	29	33	33	57	43	33	71	29	83	71	46.6
Unanimous	67	25	100	50	60	33	40	40	40	50	67	75	50	67	100	50	80	100	59.1
Favorite	43	43	43	43	57	62	29	43	57	71	29	57	43	43	43	43	86	100	51.9
Home	71	14	71	29	57	62	43	14	29	29	71	43	43	57	57	29	43	29	44.1
NN vs. Naive	🟥	⚽	⚽	⚽	⚽	🟥	⚽	⚽	🟥	🟥	🟥	⚽	⚽	⚽	⚽	⚽	🟥	🟥	🟥

- All approaches lead to volatile results

- NN tends to be not worse than ‘naive‘ 11 ⚽ 8 🟥

Since not every gameday has the same number of games due to inclusion restrictions (see slide 4), the total may deviate from the average of all gamedays.

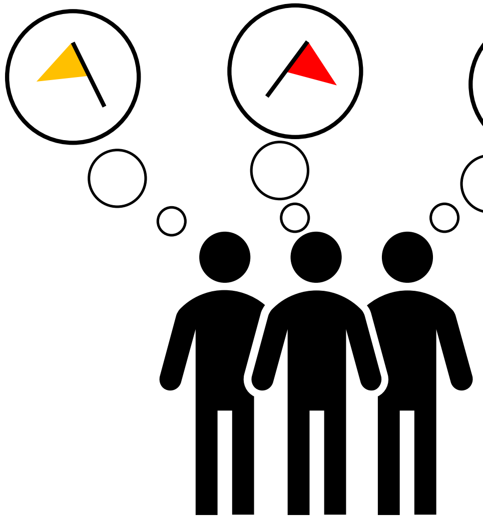
Summary

Results:

- There is some predictability in the data
- The Neural Network works best, but is not a great predictor

Use case recommendation:

- 'Naive' approaches do just fine



Code:

https://github.com/FlorianParche/ubiquitous-octo-winner/tree/master/Predicting%20the%20Bundesliga%20with%20ML