Duale Hochschule Baden-Württemberg Mannheim

Projektreport

Bundesliga Match Predictions

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser:	Max Bernauer, Philipp Dingfelder, Julius Könning
Matrikelnummer:	5763624, 8687786, 7305370
Firma:	SAP SE, Schaeffler
Modul:	Data Exploration Project
Kurs:	WWI20DSB
Dozent:	Simon Poll
Bearbeitungszeitraum:	Sommersemester 2022

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Titel "*Bundesliga Match Predictions*" selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Max Bernauer, Philipp Dingfelder, Julius Könning

Inhaltsverzeichnis

1 Zufall und Vorhersehbarkeit im Fußball

Was macht Fußball so spannend und beliebt? Sepp Herberger hat dazu eine klare Meinung: „Fußball ist deshalb spannend, weil niemand weiß, wie das Spiel ausgeht."

Dass Sepp Herberger teilweise Recht hat, bestätigen immer wieder vermeintliche Underdogs, die den klaren Favoriten in einem Spiel schlagen oder sogar düpieren können. Hierfür gibt es Statistiken, die den Effekt erklären: So fallen 47% aller Tore zufällig, also durch einen Fehler in der Verteidigung oder einen abgefälschten Schuss (Vgl. Stefan Galler, 2018). Außerdem ist jeder dritte Ballkontakt ein unvorhersehbares Ereignis. Es wird also beinahe unmöglich einen perfekten Spielzug mit anschließendem Torerfolg zu planen (Vgl. Dr. Roland Loy, 2011).

Dennoch gibt es auch Komponenten, die das Gegenteil vermuten lassen. So wissen wir aus Berechnungen, dass die Mannschaft mit dem höheren Marktwert zu 48% gewinnt, das Heimteam zu 45% oder das Team mit der besseren Tabellenplatzierung zu 44%. Inwieweit ist Fußball also etwa durch Machine Learning vorhersehbar oder überwiegt doch die Zufallskomponente? Und ist es tatsächlich möglich Anbieter wie Bet365 und Bwin durch ein mathematisches Modell zu schlagen? Diese beiden Fragen versuchen wir im Rahmen der Arbeit zu beantworten.

Das erfolgreiche Vorhersagen von Fußballspielen wäre auch mit einem enormen wirtschaftlichen Nutzen versehen. So kann man durch eine solche Prediction potenziell starken Profit bei etwaigen Wettanbietern erzielen. Andererseits kann man diese Software auch an Wettanbieter verkaufen, damit diese ihre Quoten noch effizienter und genauer berechnen können.

2 Theoretische Grundlagen

2.1 Related Work

Nach Recherche zu Vergleichsprojekten wurden einige Ideen aufgegriffen. Beim Preprocessing wurden mehrere relevante Features hinzugefügt, primär Elo und Angriffs- und Verteidigungsstärke (Vgl. Nicolai Minter, 09/12/2020). Als Möglichkeit für das Modelltraining wurde ein mathematischer Ansatz gefunden. Dieser ist mit R programmiert und schafft laut eigenen Angaben 64% Accuracy (Vgl. Doan, 15/05/2019). Für die grundlegende Idee des Business Use Case, der sich mit der Simulation von Wetten befasst, wurde eine weitere Quelle herangezogen (Vgl. Hartley, 2022).

2.2 Verwendete Technologien und Bibliotheken

Für die Datenvorbereitung werden klassische Python-Module wie Pandas, Sklearn und Matplotlib genutzt.

Bei der Modellimplementierung wurde zu Beginn ein Decision Tree Classifier genutzt. Dieser stammt aus dem Sklearn-Modul tree und erreichte keine nennenswerten Ergebnisse. Nach Verwerfen dieses Classifiers wurde sich für das Ensemble-Learning entschieden. Dabei nutzt man mehrere Algorithmen um sie zeitgleich zu vergleichen und potenziell zu vereinen. Die Algorithmen wie beispielsweise Logistic Regression oder Gaussian Naive Bayes stammen auch von Sklearn. Final wurde ein Neural Network implementiert. Dazu wurde Keras genutzt. Aus Keras wurde außerdem eine Random Parameter Search zur weiteren Optimierung angewandt. Durch Matplotlib wurden die Ergebnisse visualisiert. Außerdem wurde eine Poisson-Verteilung genutzt um den mathematischen Ansatz zu implementieren.

3 Praktische Umsetzung der Match Prediction

3.1 Umsetzung

Die gewählte Datenbasis enthält alle Bundesliga-Spiele seit der Saison 2006/2007 mit 65+ Features. Während des Preprocessings wurde die Anzahl der Features durch Entfernen irrelevanter Features deutlich reduziert. Diese Features wurden dann harmonisiert und mit einer Korrelationsmatrix auf ihre Aussagekraft geprüft. Zur Verbesserung der Aussagekraft vom Datensatz wurden die aktuelle Form der Mannschaften (Elo) sowie Angriffs- und Verteidigungswerte errechnet. Um die Aussagekraft weiter zu erhöhen wurden außerdem Features eingeführt, die einen direkten Vergleich der Mannschaften ermöglichen. Diese Features sind bspw. Differenzen der Elo und Angriffs- sowie Verteidigungswerte oder die Anzahl der Punkte und Tore, die eine Mannschaft während der letzten Spiele erzielt hat.

	DiffEloOld	DiffAttackOld	DiffDefendOld	PDiff3Matches	PDiff10Matches	PQuotAllMatches	MarketValueDiff	DirectComparisonHG	DirectComparisonAG
4886	172	0	0	0.0	7.0	1.527778	408.45	2.000000	0.800000
4887	683	0	0	2.0	8.0	1.894737	57.87	1.333333	0.666667
4888	-603	-2	-2	-2.0	-18.0	0.482759	-452.10	1.000000	1.000000
4889	230	2	-1	-1.0	7.0	1.970588	428.85	2.400000	1.200000
4890	48	0	3	5.0	2.0	1.107143	227.60	1.600000	1.400000
4891	5	1	0	4.0	1.0	1.095238	-130.75	1.000000	1.000000
4892	22	0	2	4.0	7.0	0.914894	92.35	1.600000	1.400000
4893	-28	-1	-1	-4.0	-6.0	0.584906	92.97	1.400000	1.000000
4894	357	1	1	1.0	7.0	1.279070	40.90	1.800000	1.100000
4895	-293	-1	1	3.0	-7.0	0.545455	-565.15	0.600000	3.000000

Abbildung 3.1: Datensatz.

Dadurch, dass die Zielvariablen bereits im Datensatz vorliegen handelt es sich um ein supervised Learning-Problem. Dieses wurde wie im Theorieteil erwähnt mit drei verschiedenen Methoden angegangen. Dem Ensemble Learning, einem Neural Network sowie einem mathematischen Ansatz.

3.2 Ergebnisse und Use Case-Validierung

Angefangen beim Ensemble Learning erkennt man verschiedene Accuracys je Algorithmus.

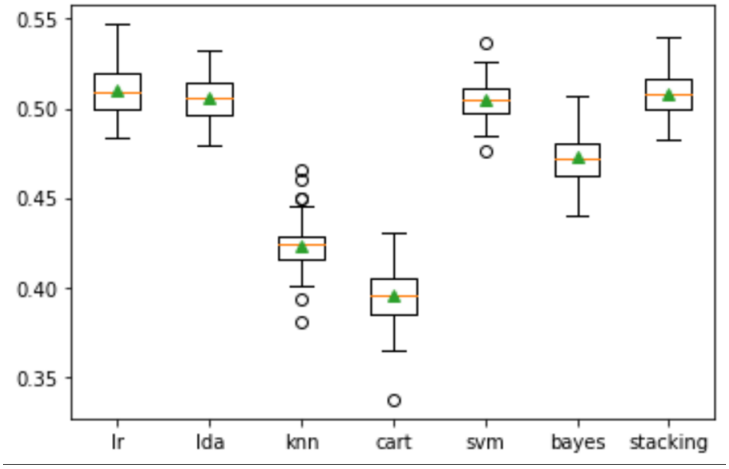

Abbildung 3.2: Ensemble Learning.

Hierbei sind die Logistic Regression und Linear Discriminant Analysis am Besten. Das Stacking Modell, welches alle Modelle vereint, schneidet ebenfalls gut ab. Alle drei liegen bei einer Accuracy von etwas über 50%. Bei Betrachtung des Neural Networks fällt auf, dass die Accuracy schon nach wenigen Epochen auf leicht über 50% konvergiert.

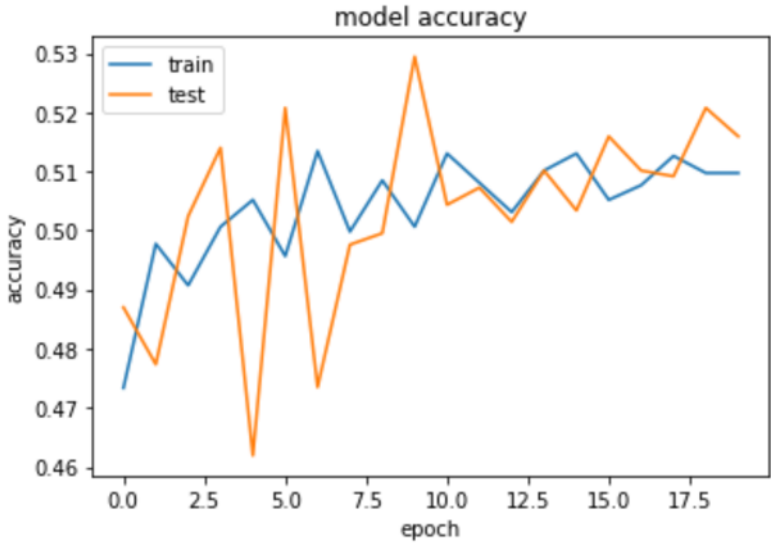

Abbildung 3.3: Neural Network.

Mithilfe einer Random Parameter Search konnte ein Maximum von 53% erreicht werden. Nach finaler Evaluierung des Modells lag es insgesamt bei ungefähr 52%. Zuletzt haben wir mit der Poisson-Verteilung des mathematischen Ansatzes eine Accuracy von etwa 46% erreicht. Die Angaben der Quelle konnten somit für unseren Fall nicht bestätigt werden.

Um zu Testen, wie gut das Modell unter Realbedingungen funktioniert wurden im Verlauf der Saison 2021/22 Wetten simuliert. Für die Simulation wurden die Predictions des Modells mit den Vorhersagen von vier verschiedenen Wettanbietern verglichen. Um die Predictions der Wettanbieter zu erhalten wurde zuerst der Durchschnitt der Odds der Wettanbieter für jedes Ereignis (Sieg Heimteam, Unentschieden, Sieg Auswärtsteam) ermittelt. Dieser Durchschnitt wurde dann in eine Prozentzahl umgewandelt und nach dem Maximum gefiltert um eine Spielvorhersage für jeden Spieltag zu erhalten. Um auf die Predictions zu setzen mussten diese einen gewissen Schwellwert (XX% Sicherheit) überschreiten. Bei Überschreitung des Schwellwerts wurde je nach Höhe der Prozentzahl ein bestimmter Betrag gewettet.

Auf die gesamte Saison 2021/22 betrachtet ergibt sich bei Wetten ab 90% Sicherheit folgende Grafik.

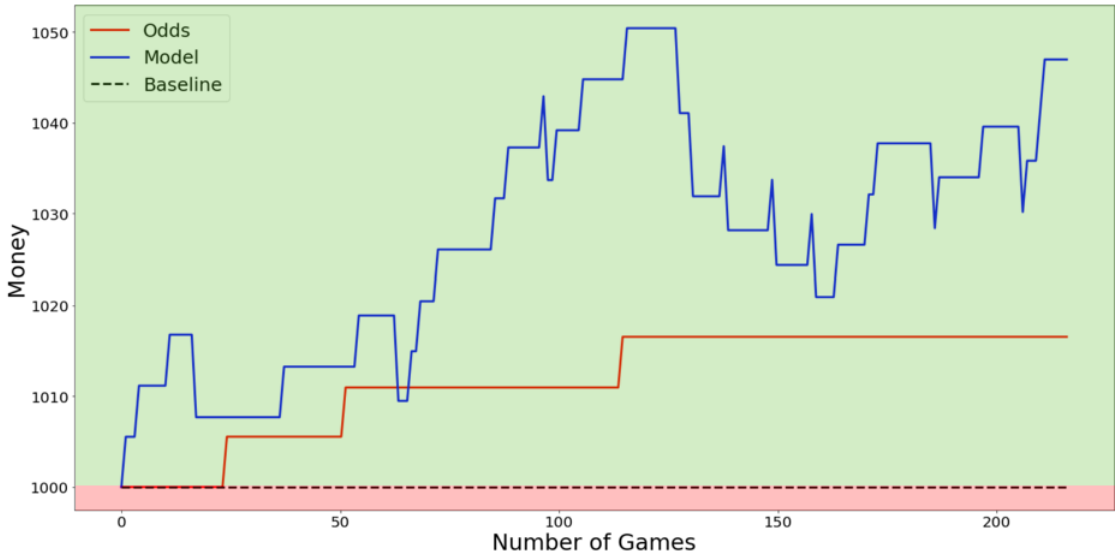

Abbildung 3.4: Betting.

Für alle Schwellenwerte lässt sich feststellen, dass es keinen signifikanten Unterschied macht, ob man unser Modell oder die Wettanbieter nutzt. Je nach Schwellenwert lässt sich mit beiden Methoden über die Saison bis zu 6% Gewinn erzielen. Auf die Bedeutung der Grafik in Bezug auf den Business Use Case wird im Fazit genauer eingegangen.

4 Fazit

Abschließend lässt sich feststellen, dass sich die Accuracy des Neural Networks im Verlauf des Projekts Schritt für Schritt durch das Optimieren der Datenbasis und des Modells erhöhen ließ. Ausschlaggebend hierfür sind vor allem das Fokussieren auf den direkten Unterschied beider Mannschaften sowie die Random Parameter Search des Modells. Final konnte eine 52% Accuracy erreicht werden.

Wie in der Grafik in Abschnitt drei von Kapitel drei gezeigt, ist es möglich mithilfe des Modells profitabel zu wetten. Vergleicht man die Profit-Kurve des Modells mit der der Wettanbieter fällt auf, dass diese im Verlauf der Saison mit zunehmender Sicherheit wesentlich weniger Wetten eingehen. Je nach Höhe des Schwellwerts können auch die Wettanbieter mehr Gewinn machen. Beide Modelle sind sich insgesamt ziemlich ähnlich im Bezug auf den generierten Gewinn. Um also langfristig die Wettanbieter gewinnbringend zu schlagen müsste das Modell weiter verbessert werden um eine höhere Accuracy zu erzielen. Ein möglicher Ansatz hierzu ist das Hinzufügen weiterer Features. Beispielsweise kann man den Marktwert der Startaufstellung betrachten oder Verletzungen von Spielern mit hinzuziehen. Außerdem erscheint eine Ausweitung der Datenbasis auf weitere Wettbewerbe sinnvoll.

Zusammenfassend lassen sich die Ergebnisse des Projekts folgendermaßen darstellen: Durch die 52% Accuracy sowie das profitable Wetten konnte gezeigt werden, dass sich Fußballergebnisse zumindest in Ansätzen algorithmisch mithilfe von Modellen beschreiben lassen. Damit ist der Use Case zum Großteil erfüllt. Es ist jedoch kritisch anzumerken, dass auch sehr viele nicht genau spezifizierbare Zufallsvariablen einen Einfluss auf den Ausgang eines Spiels haben, sodass sich nie mit 100% Genauigkeit der Ausgang eines Fußballspiels vorhersagen lässt.

A Beschreibung des Quellcodes

A.1 Ausführen des Programms

1. Klonen des Github-Repository unter dem Link https://github.com/Max280201/Data-Exploration-Project (Branch: main)
2. Öffnen der CMD und navigieren in den Git-Hub-Ordner
3. CMD-Command 'py -3.X -m venv Hoyzer' (Bash: 'python3.X -m venv HoyzerVenv') ausführen (erstellt die virtuell Environment, ersetzen von X durch die vorhandene Python-Version)
4. "HoyzerVenv/Scripts/activate.bat"in CMD ausführen (aktiviert die virtuell Environment)
5. CMD-Command "pip install -r requirements.txt" ausführen
6. Auswählen der python version der venv als Kernel; Ausführen der Python-Datei/ Notebooks anschließend möglich

A.2 Erklärung der einzelnen Bestandteile

- preprocessing_pipeline_v1.ipynb: kombiniert die csv-Dateien der Saisonspiele und die Marktwerte; außerdem Datenvorverarbeitung und Berechnung weiterer Features
- modelltraining_prod_v4.py: Modelltraining eines NN (sowohl eines, um die letzte Saison vohrerzusagen, als auch eines, um zufällige Spiele des Bereichs vorherzusagen) und kurze Evaluation des Modells
- modelltraining_test_area_v4.ipynb: enthält alle relevanten getesteten Techniken für das Modell und Alternativen (Hyperparametertuning, Ensemble-Learning, Poission-Prediction);

Anmerkung: Random Search kann zu dem Fehler "Access denied" führen, dies lässt sich durch eine geringere Anzahl an Wiederholungen beheben (https://github.com/keras-team/keras-tuner/issues/339)

- translate_betting_odds.ipynb: Auswertung und Vorverarbeitung der Wettquoten
- evaluate_model_odd_predictions_v2.ipynb: Vergleich des trainierten Modells mit Wettanbietern und Auswertung

Literatur

Doan, T. N. (15/05/2019). 'Making big bucks' with a data-driven sports betting strategy. Verfügbar 3. Juli 2022 unter https://towardsdatascience.com/making-big-bucks-with-a-data-driven-sports-betting-strategy-6c21a6869171

Dr. Roland Loy. (2011). Sportwissenschaftliche Erkenntnisse zur Taktik im Fußballsport. Verfügbar 7. Juli 2022 unter https://www.bdfl.de/images/ITK/2011/05_Loy.pdf

Hartley, L. (2022). Accurately Predicting Football with Python & SQL - Liam Hartley - Medium. *Medium*. Verfügbar 3. Juli 2022 unter https://liamjhartley.medium.com/accurately-predicting-football-with-python-sql-9353ad0e6856

Nicolai Minter. (09/12/2020). *Mit künstlicher Intelligenz zum Bundesliga-Tippkönig - ein Deepdive*. Verfügbar 3. Juli 2022 unter https://accso.de/magazin/bundesliga-tippen-mit-ki/

Stefan Galler. (2018). Fußball: Gut die Hälfte aller Tore fallen aus Zufall. *Süddeutsche Zeitung*. Verfügbar 7. Juli 2022 unter https://www.sueddeutsche.de/muenchen/fussball-tore-zufall-forschung-professor-fc-bayern-dusel-1.4008709