Methoden und Datengrundlage - Sind Regenrennen wirklich spannender?

Welche Daten liegen der Auswertung zugrunde?

Die allgemeinen Daten zu Rennen entstammen zwei von vielen Formel1-Datenbanken (`http://www.f1db.de/index.html` & `http://ergast.com/mrd/`). Die für dieses Projekt relevanten Daten umfassen Renndaten und -ergebnisse. Für die Sentiment-Analysen von Twitter- und YouTube-Kommentaren wurde jeweils die API (Application Programming Interface) verwendet. Diese Programmierschnittstellen ermöglich standardisierten Zugriff auf Daten der Internetseiten. Die Twitter-API kann über (`https://api.twitter.com/2/tweets/search/all?query=`) erreicht werden, wobei bestimmte Parameter übergeben werden müssen, je nachdem wonach gesucht wird. Die Daten beschränken sich auf Rennen zwischen 2016 und 2021 und wurden nach einem bestimmten Muster gesucht. Als Basis der Suche dient der Hashtag, der bei Twitter explizit bei der Suche angegeben werden kann. Als Hashtag wird der Name des Rennen verwendet. Dabei stellen vier Rennen eine Ausnahme dar, dessen Namen keine Ergebnisse liefern (s. Skript „twitterdataextraction.R“). Die YouTube-Daten enthalten Kommentare der „Race Highlights“ aller 84 Grand Prix von 2017 bis 2021, die nach jedem Rennen offiziell von der Formel 1 veröffentlicht werden. Nach der Durchführung der Sentiment-Analyse werden die Daten jeweils auf Rennen aggregiert. Zusätzlich werden Fan Ratings aus einem Kaggle-Datensatz verwendet (`https://www.kaggle.com/codingminds/formula-1-race-fan-ratings`). Informationen zu nassen Rennen basieren auf (`https://en.everybodywiki.com/List_of_Formula_One_wet_weather_races`) und wurden händisch sowohl auf Basis der offiziellen Formel 1 als auch einem Reddit-Post ergänzt (`https://www.reddit.com/r/formula1/comments/g0plkr/list_of_wet_weather_races_and_wins_by_driver/`).

Wichtig zu beachten ist das Ungleichgewicht zwischen nassen und trockenen Rennen. Von 2016 bis 2021 gab es lediglich 10 nasse Rennen, wobei nass als Regen während des Rennens verdeutlicht werden kann. Außerdem reichen die Datenquellen unterschiedlich weit in die Vergangenheit zurück. Bspw. reichen die Renndaten bis 2008, wohingegen die Twitter-Daten nur bis 2016 zurückreichen.

Wie wurden die Sentiments ermittelt und die Daten ausgewertet?

Die Sentiments jedes einzelnen Tweets und Kommentares werden auf Basis des CoreNLP-Paketes der Stanford Universität ermittelt (`https://stanfordnlp.github.io/CoreNLP/download.html`). Klassifiziert werden die Kommentare in drei Kategorien – positiv, negativ und neutral. Die Algorithmen arbeiten bei der Klassifizierung mit Wörtern und Wortzusammensetzungen, die mit positiven oder negativen Tendenzen verbunden werden. Für jeden Tweet oder Kommentar wird die Anzahl der positiven und negativen Begriffe ermittelt und die Häufigkeit bestimmt dann die Zuordnung zu einer der Klassen.

Zur Auswertung der Daten wird ebenfalls eine lineare Regression verwendet, um einen linearen Zusammenhang zwischen den unterschiedlichen Faktoren, u.a. auch dem Kennzeichen, ob ein Rennen nass war, und dem "Excitement Index" zu prüfen. Geprüft werden die Anzahl an Overtakes, die Anzahl der Fahrer, die das Ziel nicht erreicht haben, und, ob ein Rennen nass war. Visualisiert wird die Regression im Anschluss in einem QQ-plot.

Für die Analyse der Daten, das Sammeln der Twitter-Daten und die Erstellung der Grafiken wurde die statistische Software R (R Development Core Team, 2020) in Version 4.0.3 verwendet. Für den Zugriff auf die YouTube-API wurde die Software Python (Van Rossum and Drake, 2009). Zum Herunterladen der Twitter-Daten wurden außerdem die Packages httr (Wickham and RStudio, 2020) in Version 1.4.2, jsonlite (Ooms et al., 2020) in Version 1.7.1 und dplyr (Wickham et al., 2020b) Version 1.0.2 verwendet. Zur Visualisierung wurde das Package ggplot2 (Wickham et al., 2020a) in Version 3.3.2 und gridExtra (Auguie, 2017) in Version 2.3 in R verwendet. In Python wurden die Packages Pandas (McKinney et al., 2010), NumPy (Harris et al., 2020), textblob (Loria, 2018), Google API Client (pyt, b) und Python coreNLP (pyt, a). Zusätzlich wurden die in Python enthaltenen Funktionen re, os und itertools verwendet.

Wann kann ein Rennen als "spannend" bezeichnet werden?

Spannung ist ein dehnbarer Begriff und liegt immer im Auge des Betrachters. Um ihn trotzdem quantifizieren zu können, werden verschiedene Faktoren berücksichtigt, die aus verschiedenen Gründen einen Einfluss auf die Spannung eines Rennens haben können:

- **Verhältnis der Anzahl an positiven und negativen Kommentaren zu der Gesamtanzahl:**

Sowohl positive, als auch negative Kommentare können ein Indiz für ein spannendes Rennen sein. Die Gesamtanzahl enthält zusätzlich die neutral-klassifizierten Kommentare.

- **Verhältnis der Likes und Dislikes zu der Gesamtanzahl an Views (nur für YouTube):**

  Nicht alle Benutzer geben Kommentare ab – einige liken oder disliken das Video. Auch dies kann ein Indikator für das Interesse an einem Rennen sein. Um die Anzahl über alle Rennen vergleichbar zu machen, werden die Likes/Dislikes ins Verhältnis zur Gesamtanzahl an Views gesetzt.

- **Anzahl der Überholmanöver im Rennen:**

  Je mehr Überholmanöver in einem Rennen stattfinden, desto häufiger ändert sich die Rangliste, was das Rennen noch spannender machen könnte.

- **Anzahl der Fahrer, die das Rennen nicht beendet haben:**

  Das vorzeitige Aus eines Fahrers kann verschiedene Gründe haben – es kann technischer Natur oder durch einen Unfall herbeigeführt worden sein. Beide Gründe können einen Einfluss darauf haben, wie spannend ein Rennen ist.

- **Fan-Ratings der Rennen:**

  Fans haben zwar bereits ein allgemeines Interesse an dem Sport. Sie haben dadurch aber auch mutmaßlich höhere Ansprüche an die Rennen und geben somit eine differenziertere Meinung zu Rennen ab.

Wichtig zu beachten:

Bei der Auswertung, ob ein Rennen als „spannend“ bezeichnet wird oder nicht, werden keine absoluten Werte, wie bspw. Anzahl an Kommentaren oder Anzahl der Likes, berücksichtigt. Das ist mit der allgemein steigenden Anzahl an Kommentaren, Tweets und Likes über die Jahre zu erklären, was vor allem bei späteren Rennen Einfluss auf die Bewertung nehmen würde. Zudem werden die Faktoren einzeln betrachtet und nicht aggregiert auf einen einzigen Wert. Somit können die Rennen differenzierter verglichen und bewertet werden anhand einzelner Faktoren.

Bibliography

Python wrapper for stanford corenlp, a. URL `https://pypi.org/project/pycorenlp/`.

Google api client, b. URL `https://pypi.org/project/google-api-python-client/`.

Baptiste Auguie. *Miscellaneous Functions for "Grid" Graphics*, 2017.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. *Array programming with NumPy*, 2020.

Steven Loria. textblob documentation. 2018.

Wes McKinney et al. *Data structures for statistical computing in python*. Austin, TX, 2010.

Jeroen Ooms, Duncen Temple Lang, and Lloyd Hilaiel. *A Simple and Robust JSON Parser and Generator for R*, 2020.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Hadley Wickham and RStudio. *Tools for Working with URLs and HTTP*, 2020.

Hadley Wickham, Winston Chang, Lionel Henry, Thomas L. Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and RStudio. *Create Elegant Data Visualisations Using the Grammar of Graphics*, 2020a.

Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, and RStudio. *A Grammar of Data Manipulation*, 2020b.