Bericht DataSciece Projekt

Verbindung von Gehalt und Wahlverhalten in Frankfurt

PaSeDa

Dies ist die finale Ausarbeitung unseres Projektes für das Modul DataScience1. Untersucht wurde der Zusammenhang von Gehalt und Wahlverhalten in Frankfurt. Dazu wurden zwei Datensätze von **TODO** verwendet die Informationen von Gehalt und Wahlverhalten je Stadtteil zu Verfügung stellen. Organisiert haben war unser Projekt mit Jupyter Notebooks uns GitHub. Die gesäuberten Datensätze wurden auf verschiedene Weise untersucht. Insbesondere wurden mehrere Multi-Target-Regressoren trainiert, um festzustellen ob eine Wahlvorhersage aufgrund des Gehalts sinnvolle Ergebnisse liefert. Festgestellt haben wir, dass es einen statistisch signifikanten Zusammenhang zwischen Verteilung Gehalt eines Stadtteils und den Wählerstimmen für SPD, FDP und AFD gibt.

Inhaltsverzeichnis

1 Zielsetzung

Dass das Wahlverhalten von Deutschen durch ihre Einkommen beeinflusst würde, ist etwas dass viele schon gehört haben oder auch selber vermuten. Wir wollen diese Vermutung an Hand von Daten aus Frankfurt untersuchen.

2 Unsere Infrastruktur

Unser gesamtes Projekt ist auf `https://github.com/5yntek/DataScienceProject` zu finden. Die verwendeten Rohdaten sind unter `/data` gesammelt. **TODO** *Infrastruktur säubern und hier aufführen.* Unseren Code haben wir in Jupyter Notebooks gesammelt. Sämtlicher Code ist Python3. Nennenswerte Bibliotheken, die wir verwendet haben sind scipy (numpy, pandas, matplotlib), sklearn, seaborn und pandas-profiling.

TODO *Wie sollen wir Referenzen organisieren?*

3 Die Daten

Information über die Gehälter der Frankfurter sind leider recht rar. Die einzigen relevanten Daten haben wir in einem Datensatz zum Arbeitsmarkt gefunden. Dieser Datensatz ist auf dem Portal offenedaten.frankfurt.de

zu finden und erhebt nach eigenen Angaben Daten aus dem Jahr 2011 und 2012. Leider ist die auf der Seite angegebene Quelle veraltet, sodass wir die Daten nicht weiter verifizieren können. Dass die Daten nicht aktueller sind ist zwar Schade, sollte für unser Projekt aber kein Hindernis sein.

Mehr Daten werden über Wahlergebnisse erhoben. Alleine auf offenedaten.frankfurt gibt es diverse Datensätze zu verschiedenen Wahlen. Für uns am interessantesten erscheinen Daten zur Bundestagswahl, sodass wir uns für den Datensatz der Bundestagswahl 2017 (im weiteren BW17 genannt) entschieden haben.

Leider ist die kleinste Entität, über die Daten sowohl für Gehalt als auch Bundestagswahl erhoben wurde, das Stadtteil, so dass wir mit nur insgesamt **TODO** *(wie viele?)* Datenpunkten arbeiten.

Glücklicherweise liegen die Daten sowohl der Bundestagswahl als auch des Arbeitsmarktes in CSV- oder JSON-Dateien vor. Dies erleichtert das Einlesen der Rohdaten in `pandas.DataFrames`, der Datenstruktur in der wir intern die Daten repräsentieren und mit der wir arbeiten.

Das Einlesen der Daten findet im Notebook **TODO** unter **TODO** statt.

4 Preprocessing

Ein Blick in die Daten zeigt uns, dass viele Daten fehlen und einige für uns uninteressant sind. Auch sind die Namen der Stadtteile der beiden Datensätze nicht vollständig identisch.

4.1 Merge and clean the data

Zunächst verwerfen wir Informationen aus den Rohdaten die wir nicht gebrauchen können **TODO** Referenz, so zum Beispiel totale Werte. Die Gesamtanzahl der Wähler einer Partei ist nicht interessant, da wir nur relative Werte gebrauche können, um Stadtteile miteinander vergleichen zu können. Weitere Probleme die wir lösen mussten: Wir müssen dafür sorgen, dass numerische Werte richtig interpretiert werden. Die deutsche Variante der Darstellung von Zahlen ist untypisch und muss transformiert werden. Einige Sonderzeichen befinden sich in den Bezeichnern der Attribute, die entfernt werden müssen. Die Attributnamen sind generell unpraktisch lang und werden umbenannt. Die Stadtteile Gutleut- und Bahnhofsviertel wurden in BW17 unter *Gutleut-/Bahnhofsviertel* zusammengefasst und müssen getrennt werden. Die Anteile des Bruttoarbeitsentgelts sind nicht direkt vorhanden und müssen zunächst aus den anderen Daten errechnet werden. Nach diesen Schritten können wir die beiden Tabellen zusammenführen.

4.2 Test and verify your data quality

TODO Hat jemand eine Idee was genau wir hier schreiben sollen?

4.3 Further preprocessing

TODO

4.4 pandas-profiling

Als ersten Schritt nach dem das Säubern der Daten abgeschlossen ist, ist eine statistische Analyse der Daten. Ein Tool mit dem wir bereits gute Erfahrung gemacht haben ist pandas-profiling. Dieses Framework erzeugt mit wenigen Zeilen einen ausführlichen Bericht in Form von einem HTML-Dokument. Pandas-Profiling zeigt unter anderem fehlende Werte, Datentypen und -Bereiche, sowie Korrelation an. Letztere ist besonders interessant. Pandas-Profling liefert mehrere gut lesbare Diagramme zu Korrelationen innerhalb der Daten. Ein davon stellt die Pearson-Korrelation (r) da. Diese ist ein Maß für die lineare Korrelation zwischen zwei Variablen. Sein Wert liegt zwischen -1 und +1. -1 steht für maximale negative lineare Korrelation, 0 zeigt keine lineare Korrelation an und 1 zeigt maximale positive lineare Korrelation. Abbildung 1 stellt die Pearson-Korrelation der Daten da. Einige Zusammenhänge fallen schnell ins Auge, so hat z. B. die FDP eine sehr starke (Anti-)Korrelation zur Gehaltsgruppe 2000-bis-4000. Nicht minder stark (anti-)korreliert das Wahlergebnis der CDU mit dem der DIE LINKE.

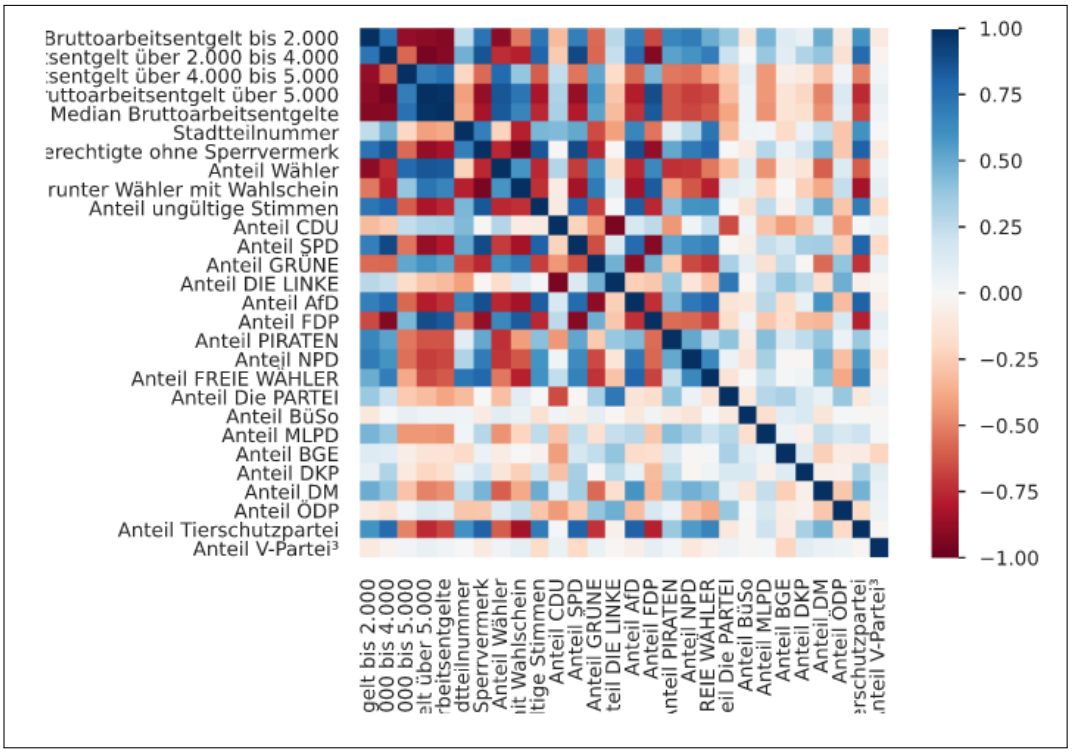

Abbildung 1: Pearson-Korrelation der Daten

5 Apply two different algorithms of the same kind

Wir haben ein interessantes Problem hier. Wir wollen weder eine Klassifizierung noch eine Regression. Unser gesuchtes Ergebnis ist die Ausprägung(Stimmanteil) verschiedener Parteien(Klassen). Ich denke One-versus-the-rest bzw. Multinomial logistic regression sollten wir auf jeden Fall anwenden.

6 Evaluate and verify the results of them!

TODO

7 Come up with a conclusion!

TODO