

Prognose des Wahlverhaltens der Stadtbezirke in Frankfurt auf Basis der Einkommensverteilung

TEAM: PaSeDa

Patrick Bonack, Sebastian Gampe, Daniel Helmke

Agenda

1. Daten und Datenanalyse
2. Daten Bearbeitung
3. Idee
4. Training
5. Ergebnisse und Verbesserung

Die Daten und Datenanalyse

Zwei Datensätze aus den offenen Daten der Stadt Frankfurt:

1. Arbeitsmarktdaten von 2010-2012:

- Arbeitslosenquote, Nebenjobdichte, Bezug von ALG II und vieles andere mehr
- Einkommensverteilung gegliedert in:
 - bis 2000, 2000-4000, 4000-5000, über 5000 und Median in Euro

2. Ergebnisse der Bundestagswahl von 2017 für alle 18 angetretene Parteien in Prozent

Datenanalyse:

- Verschieden in Python vorhanden Tools für die Korrelation einzelner Datensätze untereinander graphisch dargestellt und diskutiert.

Letztendlich für die Einkommensverteilung entschieden.

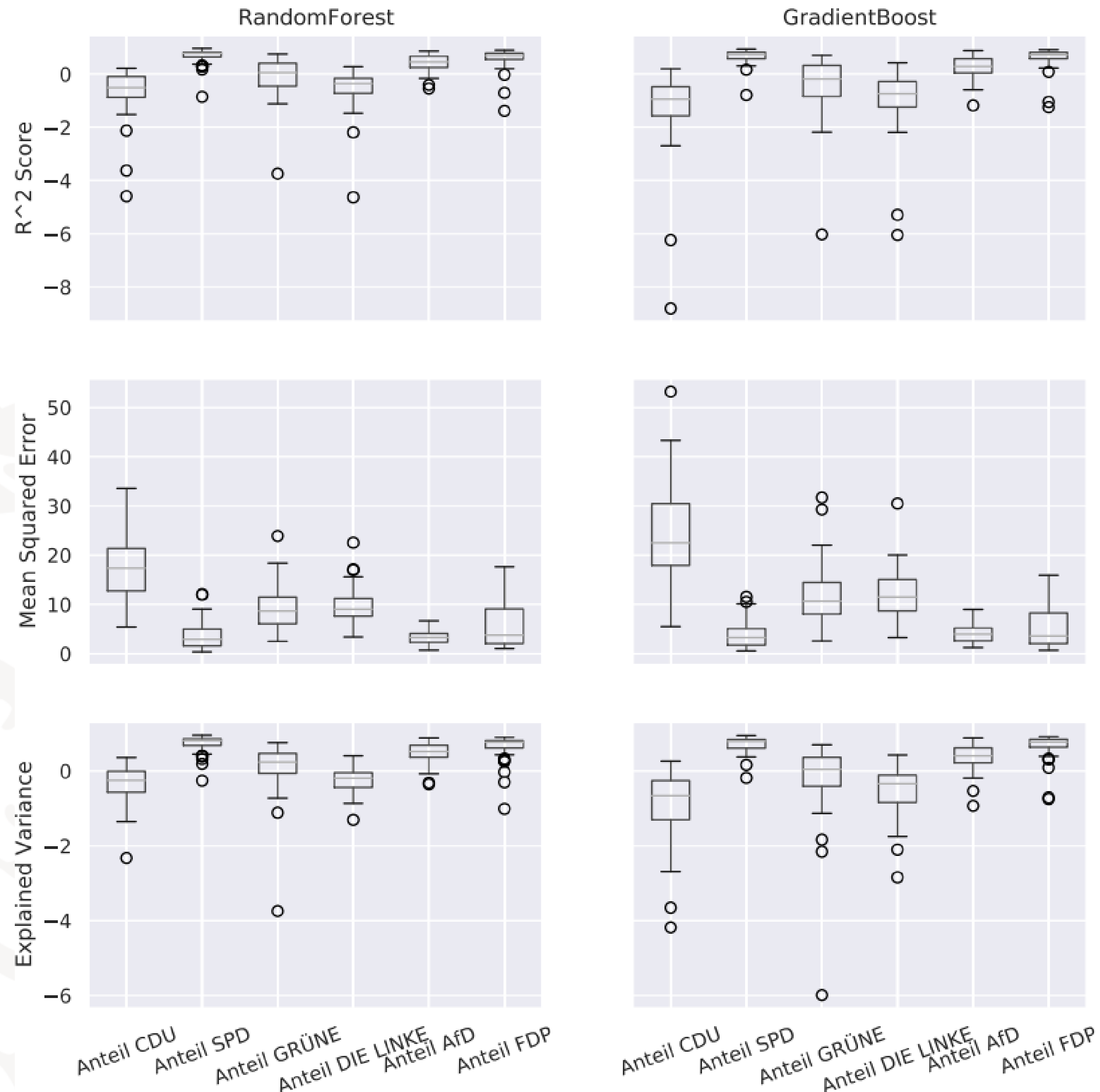
- Datenbearbeitung
 - Die Einkommensverteilung normiert
 - Zwei unvollständige Datensätze entfernt
 - Einen in beiden Datensätzen unterschiedlich repräsentierten Datensatz angepasst
 - Die beiden Datensätze über die Stadtteile zusammengeführt

Idee

- Vorhersagen der Wahlergebnisse auf Basis der Daten zur Einkommensverteilung
- Bzw. Erkennen von Zusammenhängen zwischen Wählerschaft und Einkommen
- Input: mehrere kontinuierliche Daten zur Einkommensverteilung
- Output: kontinuierliche Daten zum Anteil der Parteien am Wahlergebnis
- → Multi Target Regressionsproblem

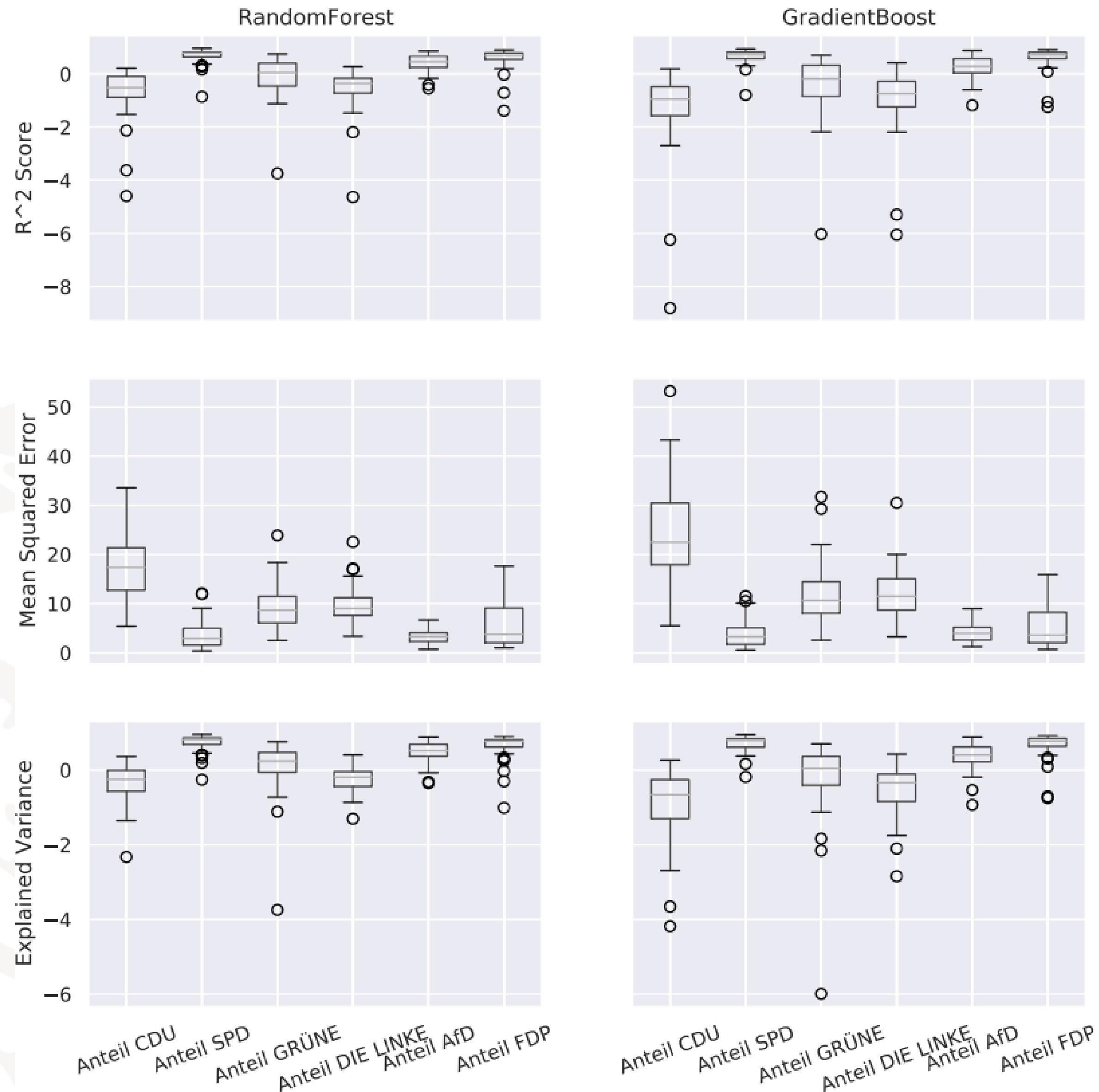
- Untersuchte Algorithmen:
 - SKLearn GradientBoostingRegressor
 - SKLearn RandomForestRegressor
- SKLearn MultiOutputRegressor
- Wiederholtes Training mit Analyse des Ergebnis
 - Einfaches Training mit 1 zu 4 Test- /Trainingsplit
 - Wiederholtes Training auf gleichen Test- /Trainingsdaten
 - Cross-Validation
 - Repeated 5-Fold-Cross-Validation aufgeschlüsselt nach Parteien

Auszug aus den Ergebnissen



- Analyse der Modelle in Abhängigkeit zu gewählten Test- Und Trainingsdaten
- Boxplot verschiedener Metriken zur Vorhersage der beiden Algorithmen Random Forest und Gradient Boost
- Ausgabe beschränkt auf CDU, SPD, Grüne, Linke, AfD und FDP
- Explained Variance, Mean Squared Error, R²
- 10 Wiederholungen 5-Fold-Cross-Validation
- Datenpunkt bezieht auf eine Instanz eines Modells mit eigenen Trainings- und Testdaten

Ergebnisse



- Allgemein sehr schlechte Metriken
→ R^2 niedrig, MSE hoch und EV zu niedrig
- Ausnahmen: SPD, FDP und AfD
- Fazit:
 - Kein gutes Modell zur Vorhersage
 - Unterschiede zwischen RandomForest und GradientBoost minimal

Verbesserungsmöglichkeiten

- Probleme:

- Datenbasis zu klein und aggregiert
- Zeitliche Differenz der einzelnen Datensätze
- Korrelation der Eingabe und Ausgabewerte

- Mit mehr Zeit:

- RandomForest als Basis
- Weitere Faktoren des Wahlverhaltens einbeziehen
- Größere Datenbasis
- Training eines Modells pro Partei

Danke für die Aufmerksamkeit!

Fragen?

