Data Science mit dem Yelp-Datensatz

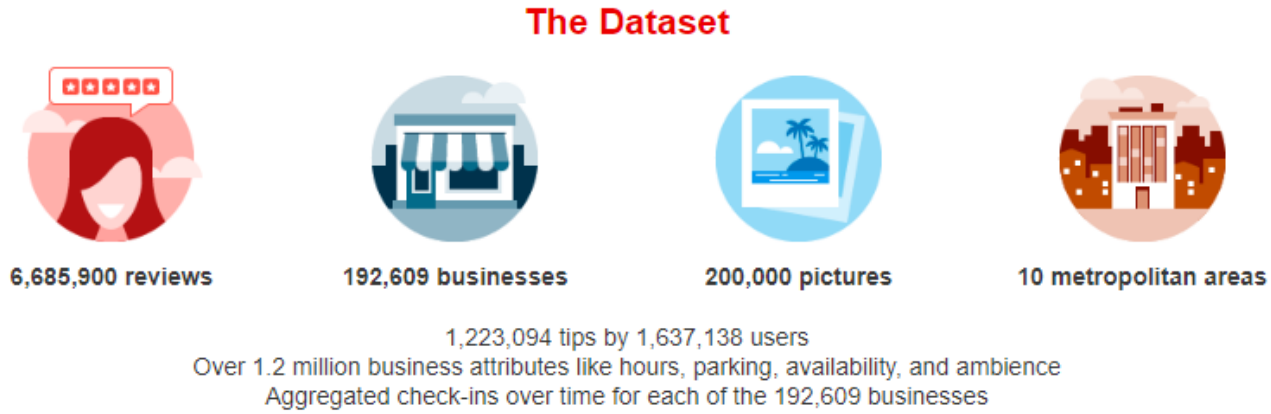
Amrit Pal Singh

May 13, 2019

Übersicht

- Der Yelp-Datensatz
- Ziel der Datenanalyse
- Säuberung und Vorbereitung der Daten
- Auswahl des Algorithmus
- Tuning der Hyperparameter
- Ergebnisse, RandomForestClassifier
- Ergebnisse, GradientBoostingClassifier
- Diskussion der Ergebnisse

Der Yelp-Datensatz



- Der Yelp-Datensatz enthält:
  - Informationen aus 192609 Geschäftsbetrieben aus 10 Metropolregionen
  - Nutzerbewertungen der Geschäftsbetriebe
  - In dieser Datenanalyse beschränke ich mich auf den Business-Datensatz

Quelle: www.yelp.com/dataset

Ziel der Datenanalyse

- Der Business-Datensatz umfasst vor allem Eigenschaften von Restaurants sowie deren Nutzerbewertungen
- Die Bewertungen sind abgerundet als ganze Zahlen im Bereich zwischen 5 (sehr gut) und 1 (schlecht) gegeben und stellen die kategorische abhängige Variable y dar
- Die Bewertung eines Restaurants wird unter anderem durch die unabhängigen Variablen (Features)
  - 'PriceRange'
  - 'GoodForGroups'
  - 'GoodForKids'

  bestimmt
- Ziel dieser Datenanalyse:
  Trainieren von Algorithmen, um die Bewertung eines Restaurants in Abhängigkeit dieser Features vorherzusagen
- Verwendete Programmiersprache: Python

Säuberung und Vorbereitung der Daten

- Entfernen von Zeilen die 'nan'- und 'none'-Werte enthalten
- Auswahl aller Features, welche die Bewertungen von Restaurants beeinflussen
- Extrahierung der drei Features 'PriceRange', 'GoodForGroups' und 'GoodForKids'
- Extrahierung der zugehörigen Bewertungen
- Transformation der kategorischen Features 'GoodForGroups' und 'GoodForKids' durch one-hot-encoding
- Erzeugung der Features-Matrix X und des Feldes y, welches die Bewertungen enthält
- Features-Matrix enthält 47472 Samples und 3 Features
- Aufteilung der Daten in Trainings- und Testdatensatz, Training $= 75\%$, Test $= 25\%$

Auswahl des Algorithmus

- Es sollen Algorithmen trainiert werden, um die Bewertung eines Restaurants in Abhängigkeit der drei Features 'PriceRange', 'GoodForGroups' und 'GoodForKids' vorherzusagen
- Bei der Bewertung handelt es sich um eine kategorische Variable mit Werten zwischen 5 und 1
- Es liegt somit ein Klassifizierungsproblem vor
- Im Rahmen dieser Datenanalyse wurden deshalb der 'RandomForestClassifier' und der 'GradientBoostingClassifier' aus der Scikit-learn Bibliothek verwendet

Tuning der Hyperparameter

- Hyperparameter RandomForestClassifier: 'n_estimators', 'max_features' und 'max_depth'
- Hyperparameter GradientBoostingClassifier: 'n_estimators', 'learning_rate' und 'max_depth'
- Zum Tuning der Hyperparamter wurde für beide Algorithmen die 'RandomizedSearchCV' mit 5-facher Kreuzvalidierung auf den Trainingsdatensatz angewendet
- Danach wurden der RandomForestClassifier bzw. der GradientBoostingClassifier mit den besten ermittelten Hyperparametern auf dem gesamten Trainingsdatensatz neu trainiert
- Anschließend wurde die erreichte Genauigkeit für Trainings- und Testdatensatz für beide Algorithmen evaluiert
- Genauigkeit = Accuracy Score = Anzahl der richtig klassifizierten Punkte aus dem Testdatensatz

Ergebnisse, RandomForestClassifier

- Die 'RandomizedSearchCV' mit 5-facher Kreuzvalidierung liefert als optimierte Hyperparameter:
  - n_estimators = 1000
  - max_features = 3
  - max_depth = 30
- Diese optimierten Hyperparameter liefern als Genauigkeiten
  - Training Accuracy Score = 0.451
  - Test Accuracy Score = 0.456

Ergebnisse, GradientBoostingClassifier

- Die 'RandomizedSearchCV' mit 5-facher Kreuzvalidierung liefert als optimierte Hyperparameter:
  - n_estimators = 1500
  - learning_rate = 0.01
  - max_depth = 3
- Diese optimierten Hyperparameter liefern als Genauigkeiten
  - Training Accuracy Score = 0.451
  - Test Accuracy Score = 0.456

Diskussion der Ergebnisse

- Sowohl der RandomForestClassifier als auch der GradientBoostingClassifier liefern einen Test Accuracy Score von 45.6 %
- Die geringe Genauigkeit ist vermutlich darauf zurückzuführen, dass zur Erklärung der Bewertungen eine zu kleine Zahl an Features verwendet wurde.
- Mögliche Lösung:
  Weitere Features wie z. B. 'BusinessAcceptsCreditCards', 'RestaurantsDelivery' oder 'RestaurantsReservations' zur Erklärung der Bewertungen verwenden
- Nachteil dieser Lösung: Viele Einträge dieser zusätzlichen Features sind vom Typ 'nan' oder 'none' und die entsprechenden Zeilen müssen somit aus dem gesamten Datensatz entfernt werden. Dies reduziert die Gesamtzahl der Trainingsdaten und kann dadurch ebenfalls zu einer Verringerung des Test Accuracy Scores führen.