Duale Hochschule Baden-Württemberg
Mannheim

Webbasierte Datenbankanwendungen

Datenanalyse über Verkehrsdaten und Unfallstatistiken in United Kingdom mithilfe von Logstash, Elasticsearch und Kibana

Studiengang Wirtschaftsinformatik

Studienrichtung Software Engineering

Modul:	Software Engineering I	
Fach:	Webbasierte Datenbankanwendungen	
Dozent:	Prof. Dr. Julian Reichwald	
Verfasser:	Daniel Pies	Matrikelnummer: 7104609
	Thomas Pötz	Matrikelnummer: 5388316
	Manuel Techert	Matrikelnummer: 7615798
Kurs:	WWI 16 SEA	

Inhaltsverzeichnis

1 Thema des Projekts

1.1 Idee

In modernen Autos gibt es immer mehr Assistenzsysteme, die den Fahrer unterstützen sollen und die Sicherheit im Straßenverkehr erhöhen sollen. Dazu zählen zum Beispiel Spurhalteassistenten, die automatisch das Fahrzeug auf die Fahrbahn zurück lenken, wenn der Fahrer die Spur verlässt. Ein weiteres System ist ein automatisches Bremssystem, wenn der Fahrer nicht genug Sicherheitsabstand zum Vordermann einhält, dieser plötzlich bremst oder ein Fußgänger unachtsam die Straße überquert. Durch solche Assistenzsysteme in Autos, könnte es sein, dass die Zahl der Verkehrsunfälle im Vergleich zu Verkehrszählungen in den letzten Jahren zurück gehen, da die Systeme Unfälle verhindern sollen.

1.2 Ziel

In dieser Arbeit sollte ein Zusammenhang zwischen Verkehrsunfällen und Daten einer Verkehrszählung in United Kingdom erarbeitet werden. Dabei sollen die Zahlen der Verkehrsunfälle und der Verkehrszählungen korreliert werden, um Abhängigkeiten und Abweichungen voneinander festzustellen.

2 Installation

2.1 Daten

Daten wurden von folgenden Seiten heruntergeladen:

- Unfallstatistiken: https://www.kaggle.com/silicon99/dft-accident-data
- Verkehrsdaten: https://www.kaggle.com/sohier/uk-traffic-counts

2.2 Notwendige Änderungen der Config-Files

Logstash akzeptiert nur absolute Pfade zu den Daten. Aus diesem Grund müssen folgende Zeilen in den Config Files durch den lokalen Pfad zum Ordner ersetzt werden:

- uk_accidents.conf:

  Zeile 3: path => “<Pfad zum Ordner>/accidents/Accidents0515.csv“

  Zeile 62: template => “<Pfad zum Ordner>/accidents/elasticsearch_template_accidents.json“

- uk_traffic.conf:

  Zeile 3: path => “<Pfad zum Ordner>/traffic/Raw-count-data-major-roads.csv“

  Zeile 60: template => “<Pfad zum Ordner>/traffic/elasticsearch_template_traffic.json“

2.3 Installation

2.3.1 Docker

Elasticsearch und Kibana werden in Docker-Containern ausgeführt. Docker kann über folgenden Link (https://www.docker.com/get-docker) heruntergeladen und installiert werden. Nachdem Docker gestartet wurde, müssen folgende Befehle auf der Kommandozeile ausgeführt werden:

- docker run -d -p 9200:9200 -p 9300:9300 -it -h elasticsearch –name elasticsearch elasticsearch
- docker run -d -p 5601:5601 -h kibana –name kibana –link elasticsearch:elasticsearch kibana

2.3.2 Logstash

Logstash läuft nicht in einem Docker-Container, sondern kann hier (https://artifacts.elastic.co/downloads/logstash/logstash-6.2.2.zip) heruntergeladen werden. Die Unfallstatistik- und Verkehrsdaten sind über die zuvor erwähnten Links herunterzuladen und in die entsprechenden Ordner \accidents bzw. \traffic zu verschieben. Anschließend können die Daten über die nachfolgenden Befehle in Elasticsearch eingefügt werden:

- <Pfad zum Ordner>\logstash-6.2.2\logstash-6.2.2\bin\logstash -f <Pfad zum Ordner>\accidents\uk_accidents.conf
- <Pfad zum Ordner>\logstash-6.2.2\logstash-6.2.2\bin\logstash -f <Pfad zum Ordner>\traffic\uk_traffic.conf

2.3.3 Kibana

Im Browser http://localhost:5601/ aufrufen:

- unter Index Patterns Index uk_accidents ohne Timefilter anlegen
- unter Index Patterns Index uk_traffic ohne Timefilter anlegen
- unter Index Patterns Index uk* ohne Timefilter anlegen
- unter Saved Objects export-kibana.json importieren

3 Vorgehensweise

3.1 Daten

Als Basis des Projekts wurden zwei Datensätze von kaggle.com in Form von CSV-Files verwendet.

1. https://www.kaggle.com/silicon99/dft-accident-data

Dieser Datensatz enthält alle erfassten Verkehrsunfälle in United Kingdom. Zu jedem Datensatz sind der exakte Zeitpunkt des Unfalls, die Längen- und Breitengrade des Unfallorts und die Anzahl der beteiligten Fahrzeuge enthalten. Es sind noch weitere Daten enthalten, die in diesem Projekt nicht verwendet wurden. Das sind zum Beispiel Infos darüber, ob der Unfall tödlich endete oder ob die Polizei dazu gerufen wurde.

2. https://www.kaggle.com/sohier/uk-traffic-counts

Dieser Datensatz enthält alle durchgeführten Verkehrszählungen in United Kingdom. Für dieses Projekt wurden wieder die Koordinaten und der Zeitpunkt der Zählung verwendet. Außerdem wurde die Anzahl der gezählten Autos verwendet.

3.2 Tools

Verwendete Tools zur Datenauswertung sind:

- Logstash in der Version 6.2.2
- Elasticsearch in der Version 5.6.8
- Kibana in der Version 5.6.8

Weitere Tools wurden zur Hilfe verwendet:

- Docker for Windows in der Version 18.03.0-ce-win59
- GitHub Desktop in der Version 1.1.1

3.3 Vorgehensweise

1. Sowohl für Elasticsearch als auch für Kibana wurde jeweils ein Docker Container installiert, die miteinander verknüpft wurden, um eine Kommunikation zwischen den Tools zu ermöglichen. Logstash hingegen wurde lokal auf dem Rechner verwendet.
2. Die heruntergeladenen Daten in Form von CSV-Files sollten mithilfe von Logstash in Elasticsearch geladen werden. Dazu wurden zwei Logstash-Config-Files geschrieben, die genau beschreiben, wie die CSVs aufgebaut sind. Es wurde angegeben durch welches Zeichen die Spalten getrennt sind, welche Spalten vorhanden sind und welchen Dateityp die Spaten haben. Zusätzlich mussten Besonderheiten angegeben werden, z. B. in welcher Form das Datum vorliegt und welche Integer-Werte die Längen- und Breitengrade sind, um einen Geo_Point Datentyp anzulegen.
3. Nachdem alle Daten in Elasticsearch vorhanden waren, konnten in Kibana die Indexe dazu angelegt werden, sowie ein gemeinsamer Index, der die Daten aus beiden CSVs beinhaltet. Danach konnten die Daten verwendet werden, um grafische Auswertungen durchzuführen.
4. Grafisch wurden zum einen zwei Heatmaps erstellt, die zeigen, wo in United Kingdom mehr und wo weniger Unfälle passiert sind, bzw. wo mehr oder weniger Verkehr ist. Außerdem wurden Graphen angelegt, die einen Verlauf von Verkehr und Unfällen über die Jahre hinweg darstellt. Für alle Auswertungen zu den Unfällen wurde die Anzahl der beteiligten Fahrzeuge verwendet, analog wurden für alle Auswertungen zum Verkehrsaufkommen die Anzahl der gezählten Autos addiert, um Unfälle und Verkehr auf einer gemeinsamen Basis vergleichen zu können.

4 Probleme

4.1 Probleme

Im Laufe des Projekts sind mehrere kleiner Probleme aufgetreten, die sofort gelöst wurden. Jedoch gab es auch folgende drei größeren Probleme:

1. Das Konvertieren von Längen- und Breitenkoordinaten aus Float-Werten in den 'Geo_Point' Datentyp.
2. Das Konvertieren von Strings in den 'Date' Typ.
3. Das Verwenden des Datum aus zwei Verschiedenen Indexen für eine gemeinsame Auswertung.

4.2 Problem: Datentyp 'geopoint'

Um mit Kibana Heatmaps zu erzeugen, die auf einer Landkarte anzeigen sollen, wo mehr und wo weniger Unfälle passiert sind, müssen die Koordinaten des Punktes in einem besonderen 'Geo_Point' Datentyp vorliegen. In beiden Datensätzen war dies natürlich noch nicht der Fall. Die Koordinaten lagen nur in Form von zwei Floatwert Longitude und Latitude vor.

4.2.1 Lösung:

Es wurde eine zusätzliche Konstruktion erstellt, die ein neues Feld erstellt. Die Werte von Longitude und Latitude wurden dann in zwei Subfelder des neuen Felds eingefügt. Danach wurde diese Konstruktion in den Geo_Point Typ konvertiert.

```
mutate {
 add_field => { "[geo_location_accidents][location][lat]"
                                  => "%{Latitude}"
                "[geo_location_accidents][location][lon]"
                                  => "%{Longitude}"
 }
}
```

4.3 Problem: Datentyp 'date'

Die Datumsfelder lagen in den CSVs als Strings vor. Sie müssen aber um mit Kibana Histogramme zu erstellen im Datentyp 'Date' vorliegen.

4.3.1 Lösung:

Man kann Strings in den Logstashfiles in ein Datum konvertieren, in dem man angibt, wie das vorhanden Datum im String aussieht. Dazu verwendet man Pattern wie zum Beispiel dd-MM-yyyy.

```
date {
 match => [ "Date", "dd/MM/yyyy" ]
 target => "Date"
}
```

4.4 Problem: Zwei Indexe

Das Datum war in zwei Verschiedenen Indexen vorhanden, sollte aber für eine gemeinsame Auswertung verwendet werden. Zusätzlich war das Datum in beiden Indexen in zwei Verschiedenen Schreibweisen angegeben. Es musste aber für eine gemeinsame Auswertung in den gleichen Typ konvertiert werden.

4.4.1 Lösung:

In beiden CSVs wurde jeweils das Datumsfeld geklont und ein neuer, in beiden Dateien gleicher Name vergeben. Wenn sowohl der Name eines Feldes als auch der Typ gleich sind, werden die zwei eigentlich unterschiedlichen Felder in einem Kibanaindex als ein Feld betrachtet und kann als Basis eines Diagramms dienen.

5 Fazit

5.1 Fazit

Es ist ein Rückgang des Verkehrsaufkommens und der Unfälle zu erkennen. Allerdings fällt die Anzahl der Unfälle im Verhältnis zum Verkehrsaufkommen nicht signifikant stärker. Somit ist entgegen der Annahme, kein zusätzlicher Rückgang der Unfälle durch Assistenzsysteme wie zum Beispiel Rückfahrkamera oder Spurhalteassistent zu erkennen. Allerdings hat sich bei der Auswertung eine andere Auffälligkeit gezeigt. In den Jahren 2009 und 2010 ist das Verkehrsaufkommen außergewöhnlich hoch.

5.2 Reflexion - Negatives

- Zeitaufwand wegen ungeplanter Hürden unterschätzt
- Es dauert lange, bis man Ergebnisse sieht, Durchhaltevermögen gefordert
- Entscheidung zwischen Szenarien – viele vorhanden, man ist jedoch von Datensätzen abhängig

5.3 Reflexion - Positives

- Zusammenarbeit untereinander
- Aufteilung und Arbeitsmoral
- Freiraum bei Wahl des Szenarios
- Freiraum bei Wahl der Tools und hohe Selbstorganisation
- Viel gelernt – Arbeit mit Elastic Stack und Docker
- Insgesamt gelungenes Projekt