

Semestrálna práca BI-BIG

Poškodené cestné komunikácie v západnom Slovensku
k 1.1.2020

Matej Šutý

B201, 2020



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Register

1. Úvod

- 1.1. Predstavenie témy
- 1.2. Použité technológie

2. Dáta

- 2.1. Cesty – podrobný zoznam úsekov
- 2.2. Pasport plochy
- 2.3. Linescan: poruchy – zamerané úseky ciest
- 2.4. Critical Roads
- 2.5. Detailed Roads

3. Postup

- 3.1. Docker
- 3.2. Apache Spark
- 3.3. Logstash
- 3.4. Elastic Search
 - 3.4.1. *Query*
 - 3.4.2. *Vizualizácie*

4. Záver

- 4.1. Zhodnotenie
- 4.2. Zdroje



1. Úvod

1.1. Predstavenie témy

Na predmete BI-BIG som dostal možnosť oboznámiť sa s prácou s novými databázovými technológiami a prístupom ku spracovaniu tzv. *big data*. Big data sú spoločným názvom pre súbory dát, ktoré majú špecifické vlastnosti (3V), ktoré znemožňujú ich spracovanie bežnými softwarovými riešeniami v rozumnom čase.

Vlastnosti 3V označujú:

- **Velocity**
Dáta prichádzajú v reálnom čase a potrebujeme ich ihneď spracovať.
- **Volume**
Objem dát narastá exponenciálne. To si vyžaduje dynamické a rýchle škálovanie.
- **Variability**
Okrem štrukturovaných dát spracujeme aj dáta z multimediálnych zdrojov, rôznych senzorov, atď.

Na internete som hľadal zaujímavý zdroj dát, ktorému budem rozumieť a získané informácie budem schopný interpretovať. Prehľadal som stránky otvorených dát [Českej republiky](#), [Slovenskej republiky](#) a [Európskej únie](#). Pochádzam z Bratislavy, hlavného mesta Slovenska a preto ma zaujal dataset snímania stavu cestných komunikácií v západnom Slovensku zo [Slovenskej správy ciest](#).

1.2. Použité technológie

V tejto práci som použil tzv. ELK stack. Obsahuje tri hlavné komponenty:

- **ElasticSearch**
 - indexácia a vyhľadávanie v dátach
- **Logstash**
 - spracovanie a sprístupnenie dát z rozličných zdrojov
- **Kibana**
 - vizualizácia a objavovanie znalostí v dátach

Viac informácií na <https://www.elastic.co/elastic-stack>.



2. Dáta

Všetky dáta sú v štrukturovanom otvorenom formáte csv, ktorý spĺňa požiadavky 3 [hviezdičky otvorených dát](#). V implementácii som z praktických dôvodov zmenil názvy niektorých stĺpcov a identifikátorov kvôli nekompatibiliti diakritiky - názvy sú naďalej reprezentatívne a nedošlo k obsahovej zmene. Ak stĺpec nemá napísaný dátový typ, jedná sa o *keyword* - *string*. Po preskúmaní datasetu som zistil, že takmer každý stĺpec (okrem usekID a hodnôt UCI) občas neobsahuje hodnoty - počet chýbajúcich údajov je veľmi malý.

2.1. Cesty – podrobný zoznam úsekov

Dataset obsahuje stĺpec **usekID** (*integer*), ktorý jednoznačne definuje úsek cestnej komunikácie (ďalej CK). Pomocou neho som dáta spájal podobne ako v relačných databázach. Pre mňa najdôležitejšie boli identifikátory ich dôležitosti v systéme CK z pohľadu Slovenska, susedných štátov a celého kontinentu. Menovite sú to:

- **Tah_E**
Medzinárodná sieť „E“ - sieť cestných komunikácií v Európe, očíslovaná od E1 až vyššie
- **Tah_TEM**
TEM (Trans-European North-South Motorway Project) – projekt regionálnej spolupráce krajín južnej, východnej a juhovýchodnej Európy na poli dopravnej infraštruktúry
- **Tah_TENT**
TEN-T (Trans-European Transport Networks)-Pan-Európske koridory, súčasť širšieho projektu zahŕňajúceho popri cestných, železničných a vzdušných sieťach aj telekomunikačné a energetické siete.
- **Typ úseku**
 - **normálny** – úsek mimo križovatky, nazývaný aj medzikrižovatkový, reprezentujúci hlavný jazdný pás v príslušnom smere cesty
 - **lúč** – úsek v križovatke, prevádzajúci cestu cez križovatku, tvoriaci hlavný jazdný pás v príslušnom smere cesty
 - **vetva** – úsek v križovatke prepájajúci úseky križujúcich sa ciest, spravidla tá časť cesty v oblasti križovatky, ktorá nezodpovedá jej štandardnému medzikrižovatkovému profilu
 - **privádzač** – úsek klasifikovaný ako križovatková vetva spravidla kratší ako 500 m, ktorý privádza dopravu z cesty s neobmedzeným prístupom na cestu s obmedzeným prístupom.
 - **medzikrižovatkový** - pozri normálny



Trieda cesty(string)	Číslo cesty(integer)	ID_Useku(integer)	Počiatočný uzol(string)	Koncový uzol(string)
cesta I. triedy	2	349	3424A00501	3424A00504
cesta I. triedy	2	271728	3424A00504	3424A00506
cesta I. triedy	2	268920	3424A00504	3424A26500
cesta I. triedy	2	259541	3424A26500	3424A53100
cesta I. triedy	2	980996	3424A53100	3424A26600

Typ úseku(string)	Smer úseku(string)	Evidenčná dĺžka - m(integer)	Poradie(integer)	Stanič. Začiatku kilometrovnikové(float)
vetva	neaplikovateľný , nepoužiteľný	18	1	
vetva	neaplikovateľný , nepoužiteľný	26	2	
medzikrižovatko vý	oba smery trasy CK	145	3	0,037
medzikrižovatko vý	oba smery trasy CK	98	4	0,182
medzikrižovatko vý	oba smery trasy CK	117	5	0,28

Stanič. koncakilometrovnikové(float)	Stanič. Začiatku kumulatívne(float)	Stanič. Konca kumulatívne(float)	Dopravný smer(string)	Legislativný stav(string)
			smer dopravy súhlasný s orientáciou úseku (jednosmerka)	neuvedený
			smer dopravy súhlasný s orientáciou úseku (jednosmerka)	neuvedený



0,182	0	0,145	oba dopravné smery (obojsmerka)	neuvedený
0,28	0,145	0,243	oba dopravné smery (obojsmerka)	neuvedený
0,397	0,243	0,36	oba dopravné smery (obojsmerka)	neuvedený

Kraj(string)	Okres(string)	Ulica(string)	Funkčná klasifikácia(string)	StavebnáKategoria(string)
Trnavský kraj	Skalica		nadregionálny	Hlavná miestna cesta
Trnavský kraj	Skalica		nadregionálny	Hlavná miestna cesta
Trnavský kraj	Skalica		nadregionálny	Hlavná miestna cesta
Trnavský kraj	Skalica		nadregionálny	Hlavná miestna cesta
Trnavský kraj	Skalica		nadregionálny	Hlavná miestna cesta

Rozhodnutie(string)	TypSpravcu_Skratka(string)	Spravca_Skratka(string)	Vlastnik_Skratka(string)	prejazdovaDlžka(integer)
	SSC	SSC - IVSC BA	STAT	23
	SSC	SSC - IVSC BA	STAT	33
	SSC	SSC - IVSC BA	STAT	145
	SSC	SSC - IVSC BA	STAT	98
	SSC	SSC - IVSC BA	STAT	117



Tah_E(string)	Tah_TEM(string)	Tah_TENT(string)	Popis zac. Uzla(string)	Popis konc. Uzla(string)
Nie	Nie	Nie	křiž. I/2, I/51, Holíč, okr.Skalica	křiž. I/2, I/51, Holíč, okr.Skalica
Nie	Nie	Nie	zač. CK I/2, križ. I/2, Holíč, okr.Skalica	zač. CK I/2, križ. I/2, Holíč, okr.Skalica
Nie	Nie	Nie	zač. CK I/2, križ. I/2, Holíč, okr.Skalica	zač. CK I/2, križ. I/2, Holíč, okr.Skalica
Nie	Nie	Nie	křiž. I/2, MK, Holíč, okr.Skalica	křiž. I/2, MK, Holíč, okr.Skalica
Nie	Nie	Nie	křiž. I/2, Holíč, okr.Skalica	křiž. I/2, Holíč, okr.Skalica



2.2. Pasport plochy

Súbor poskytuje detailný popis každého úseku ciest:

- **typKrytuVozovky**
udáva typ krytu vozovky, môže nadobúdať nasledovné hodnoty:
1 – dláždený
2 – betónový
4 – štrkový
5 – drevený
6 – valcovaná na náhradné bitúmenové spojivá
7 – bitúmenová
T – nespevnená
- **šírka vozovky (float)**
udáva hodnotu šírky vozovky v metroch s presnosťou na decimeter
- **dĺžka vozovky (m) (float)**
je dĺžka VŠETKÝCH úsekov, ktoré prislúchajú príslušnej ceste, t.j. vrátane úsekov v protismere cesty pri smerovo rozdelených cestách a vrátane križovatkových vetiev.
- **Plocha vozovky (float)**
plocha v najužšom mieste vozovky

Trieda CK (string)	Číslo CK(integer)	úsek ID(integer)	Poč. Uzol(string)	Konc uzol(string)
cesta I. triedy	2	268920	3424A00504	3424A26500
cesta I. triedy	2	268920	3424A00504	3424A26500
cesta I. triedy	2	268920	3424A00504	3424A26500
cesta I. triedy	2	349	3424A00501	3424A00504
cesta I. triedy	2	349	3424A00501	3424A00504



Konc uzol(string)	Typ úseku(string)	smer úseku(string)	evid. Dĺžka úseku(integer)	úsekove stanic. zac. javu)m(integer)
3424A26500	medzikrižovatko vý	oba smery trasy CK	145	0
3424A26500	medzikrižovatko vý	oba smery trasy CK	145	15
3424A26500	medzikrižovatko vý	oba smery trasy CK	145	87
3424A00504	vetva	neaplikovateľný , nepoužiteľný	18	0
3424A00504	vetva	neaplikovateľný , nepoužiteľný	0	0

KM stanič. konc. javu(integer)	šírka vozovky(float)	typKrytuVozovky(string)	Umiestnenie Korekcie(string)	Korekcia - príčina(string)	Plocha korekcie(float)
145	12	bitúmenová			
145	12	bitúmenová			
145	12	bitúmenová			
	5,5	bitúmenová			
		bitúmenová	S	križovatková	9,69

Plocha vozovky bez korekcie(integer)	Plocha vozovky(integer)	Poradie(integer)	Kraj(string)	Okres(string)	Správca(string)	Typ správcu(string)	Vlastník(string)
180	180	0	Trnavský kraj	Skalica	SSC - IVSC BA	SSC	STAT
864	864	0	Trnavský kraj	Skalica	SSC - IVSC BA	SSC	STAT
696	696	0	Trnavský kraj	Skalica	SSC - IVSC BA	SSC	STAT
99	99	1	Trnavský	Skalica	SSC -	SSC	STAT



			kraj		IVSC BA		
	9,69	1	Trnavský kraj	Skalica	SSC - IVSC BA	SSC	STAT



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

2.3. Linescan: poruchy – zamerané úseky ciest

Na moje potreby najdôležitejší dataset - obsahuje detaily o poškodení jednotlivých úsekov CK, ich kategorizáciu atď. Najdôležitejšie stĺpce:

- **Krok merania / m** (*float*)
vyhodnocovací interval
- **UciMax** (*float*)
maximálna hodnota indexu trhlín (Unified Cracks Index) na CK
- **UciPriem** (*float*)
priemerná hodnota indexu trhlín (Unified Cracks Index) na CK
- **DatumZberu** (*date : d.m.yyyy*)
- **Spravca_Skratka**
zodpovedný správca CK
- **Vlastnik_Skratka**
vlastník CK
- **Okres**
- **Kraj**

UCI Index:

UCI	Klasifikačný stupeň	Hodnotenia
0,0 –5,0	1	výborný stav
5,1 –10,0	2	veľmi dobrý stav
10,1 –19,0	3	vyhovujúci stav
19,1 –30,0	4	nevyhovujúci stav
> 30	5	havarijný stav



TriedaCK (string)	cisloCk (integer)	usek (integer)	pociatocnyUzo l (string)	koncovyUzol (string)
cesta I. triedy	2	7589	4424A02802	4424A52301
cesta I. triedy	2	7589	4424A02802	4424A52301
cesta I. triedy	2	7589	4424A02802	4424A52301
cesta I. triedy	2	7589	4424A02802	4424A52301
cesta I. triedy	2	7589	4424A02802	4424A52301

UsekoveStanic eZaciatku (integer)	UsekoveStanic enieKonca (integer)	Krok merania / m (integer)	UciMax (float)	UciPriem (float)
0	20	20	11,4	6,6
20	40	20	13,3	7
40	60	20	11,4	6,6
60	80	20	8,9	5
80	100	20	4,1	2,6

Umiestnenie (string)	TypParovania (string)	DatumZberu (date)	KMStanicieniez aciatku_km (float)	KMStanicienie Konca_km (float)
Jazdný pás v smere CK	Jeden priebežný pruh	09.07.2019	88,34	88,36
Jazdný pás v smere CK	Jeden priebežný pruh	09.07.2019	88,36	88,38
Jazdný pás v smere CK	Jeden priebežný pruh	09.07.2019	88,38	88,4
Jazdný pás v smere CK	Jeden priebežný pruh	09.07.2019	88,4	88,42
Jazdný pás v smere CK	Jeden priebežný pruh	09.07.2019	88,42	88,44



KumulStanice nieZaciatku_m (float)	KumulStanice nieKonca_m (float)	typMerania (string)	TypUseku (string)	SmerUseku (string)
86,561	86,581	Bežné meranie	medzikrižovatko vý	v smere trasy CK
86,581	86,601	Bežné meranie	medzikrižovatko vý	v smere trasy CK
86,601	86,621	Bežné meranie	medzikrižovatko vý	v smere trasy CK
86,621	86,641	Bežné meranie	medzikrižovatko vý	v smere trasy CK
86,641	86,661	Bežné meranie	medzikrižovatko vý	v smere trasy CK

Poradie (integer)	EvidencnaD lзкаUseku_ m (integer)	Spravca_Sk ratka (string)	Vlastnik_Sk ratka (string)	Okres (string)	Kraj (string)
435	235	MAG-BA	BA	Bratislava V	Bratislavský kraj
435	235	MAG-BA	BA	Bratislava V	Bratislavský kraj
435	235	MAG-BA	BA	Bratislava V	Bratislavský kraj
435	235	MAG-BA	BA	Bratislava V	Bratislavský kraj
435	235	MAG-BA	BA	Bratislava V	Bratislavský kraj



2.4. Critical Roads

Agregovaný dataset, ktorý vychádza z Linescan - poruchy datasetu. Pomocou UDF funkcie a agregácie som priradil hodnotenie úseku podľa UCI. Filtrovaním som vybral iba vážna poškodené (kl. stupeň >3) CK.

2.5. Detailed Roads

Agregovaný dataset, ktorý spája iba dôležité stĺpce o CK z Cesty – podrobný zoznam úsekov a Pasport plochy.



3. Postup

3.1. Docker

Používam virtualizáciu pomocou Docker kontajnerov. Príkazy na spustenie sa nachádzajú v [repozitári](#) v README inštrukciách.

3.2. Apache Spark

<pre>val hole0 = spark.read.format("csv") .option("sep", ";") .option("inferSchema", "true") .option("header", "true") .option("encoding", "cp1250") .load("ptp_linescanparovane-2019.csv") val hole = hole0.withColumnRenamed("usek", "usekID").withColumnRenamed("cisloCk", "Číslo CK")</pre>	použil som scala api od sparku na načítanie do dataframu z csv. Option(infersche ma, true) nastavila datove typy v DF automaticky. Zmenil som nazvy niektorých stĺpcov
<pre>val sec0 = spark.read.format("csv") .option("sep", ";") .option("inferSchema", "true") .option("header", "true") .option("encoding", "cp1250") .load("rs_utriedeneuseky_2020-01-01.csv") val sec = sec0.withColumnRenamed("Trieda cesty", "TriedaCK").withColumnRenamed("Číslo cesty", "Číslo CK").withColumnRenamed("ID_Useku", "usekID")</pre>	- -
<pre>val road0 = spark.read.format("csv") .option("sep", ";") .option("inferSchema", "true") .option("header", "true") .option("encoding", "cp1250") .load("pasportplochy_sr_2020-01-01.csv") val road = road0.withColumnRenamed("Trieda CK", "TriedaCK").withColumnRenamed("úsek ID", "usekID")</pre>	- -



<pre> val usefulColumnsSec = List("usekID", "TriedaCK", "Číslo CK", "Typ úseku", "Kraj", "Okres", "Dopravný smer", "Funkčná klasifikácia", "StavebnaKategoria", "prejazdovaDlžka", "Spravca_Skratka", "Vlastnik_Skratka") val usefulColumnsRoad = List("usekID", "dĺžka vozovky (m)", "Šírka vozovky", "typKrytuVozovky") val usefulColumnsHole = List("usekID", "UsekoveStaniceZaciatku", "UsekoveStanicenienieKonca", "UciMax", "UciPriem", "DatumZberu", "Krok merania / m") val usefulSec = sec.select(usefulColumnsSec.head, usefulColumnsSec.tail: _*) val usefulRoad = road.select(usefulColumnsRoad.head, usefulColumnsRoad.tail: _*) var usefulHole = hole.select(usefulColumnsHole.head, usefulColumnsHole.tail: _*).filter(\$"UciPriem" > 19) </pre>	<p>Vybral som si niektore stĺpce, ktoré ma zaujímajú a ostatné som zahodil. Využil som scala notáciu pomocou magických <code>_*</code>, ktoré som našiel na internete. Pomocou filtra som vybral iba tie záznamy o poškodení ciest, ktoré popisujú úseky s UCI poškodením > 19</p>
<pre> import org.apache.spark.sql.functions.udf val uciToClass = (uci: Int) => uci match { case it if 0 until 5 contains it => 1 case it if 5 until 10 contains it => 2 case it if 10 until 19 contains it => 3 case it if 19 until 30 contains it => 4 case _ => 5 } val classToRating = (uciclass: Int) => uciclass match { case 1 => "výborný stav" case 2 => "veľmi dobrý stav" case 3 => "vyhovujúci stav" case 4 => "nevyhovujúci stav" case 5 => </pre>	<p>Vytvoril som udf objekt, ktorý prevádza float value UCI indexu na klasifikáciu poškodenia. Táto klasifikácia je v int intervale 0 až 5. Podľa pôvodnej tabuľky zo zdroja som interpretoval každú klasifikáciu slovnou pomocou udf <i>classToRating</i>.</p>



<pre>"havarijný stav" case _ => null }</pre>	
<pre>// vytvoriť nový dataset, ktorý bude agregovať data z jedného pôvodného datasetu val uciclass = udf(uciToClass) val rating = udf(classToRating) val critical = usefulHole.groupBy("usekID", "DatumZberu").agg(functions.max("UciPriem").as("MaxUciPriemer"), functions.avg("UciPriem").as("UciPriemer")).withCo lumn("Rating", rating(uciclass(\$"UciPriemer"))).sort("DatumZberu") critical.show(1000)</pre>	<p>použil som wide transformáciu groupby - všetky záznamy o usekoch som spojil podľa useky a vytvoril som ucelený pohľad na usek, ktorý nezáleží na jednotlivých častiach. Tento usek som následne ohodnotil cez udf objekty <i>uciclass</i>, <i>rating</i>. Pridal som stĺpec cez agregáciu vyberu najväčšieho poškodenia v danom useku <i>MaxUciPriemer</i> a priemerneho poškodenia celého useku pomocou <i>avg</i> všetkých častí useku.</p>
<pre>//vytvoriť nový dataset, ktorý bude agregovať data ze dvou pôvodných datasetů najednou val detailedRoad = usefulSec.join(usefulRoad, usingColumns=Seq("usekID"), joinType = "right") detailedRoad.show()</pre>	<p>Dataseťy <i>usefulSec</i> a <i>usefulRoad</i> mali rôzne stĺpce ale niektoré mali rovnaké. Spojil som ich cez join a získal som tak wide transformáciou nový väčší dataset, ktorý obsahoval všetky zaujímavé stĺpce.</p>



<pre> secCriticalRating.repartition(1).write .format("csv") .option("sep", ";") .option("header", "false") .save("secCriticalRating") detailedRoad.repartition(1).write .format("csv") .option("sep", ";") .option("header", "false") .save("detailedRoad") </pre>	uložit dataset na disk - repartition(1) ho uloží na do jedného súboru a nevytvorí sa množstvo čiastkových súborov.
--	--

Začal som načítaním súborov do Spark DataFrame. Vstupné súbory majú Windows-1250 encoding, hlavičky a oddelovač “;”.

Pre ľahšiu prácu s dátami som premenoval niektoré stĺpce a vyhol som sa tak používaniu diakritiky a ľahšiemu spájaniu pomocou indexu *usekID*.

Následne som si vybral iba niektoré stĺpce s ktorými som pracoval - zber dát o cestách je veľmi detailný a kvôli prehľadnosti som sa sústredil iba na niektoré údaje.

- dataframe **sec** som zredukoval na **usefulSec**
- dataframe **road** som zredukoval na **usefulRoad**
- dataframe **hole** som zredukoval na **usefulHole**

V dataframe snímania poškodení ciest **usefulHole** som filtroval iba kriticky poškodené cesty ($UCI > 19$), jednotlivé časti CK som spojil cez groupby pomocou *usekID* a pomocou UDF funkcií som priradil každému úseku *Hodnotenie* poškodenia. Pre celý úsek som vyráťal priemerné poškodenie *UciPriemer* a maximálne priemerné poškodenie jednej časti CK v celom úseku *MaxUciPriemer*. Takto vznikol dataframe **critical**.

Dataframey **usefulSec** a **usefulRoad** som spojil a získal tak dataframe **detailedRoad**, ktorý v sebe obsahuje mnou požadované stĺpce *usekID*, *TriedaCK*, *Číslo CK*, *Typ úseku*, *Kraj*, *Okres*, *Dopravný smer*, *Funkčná klasifikácia*, *StavebnáKategoria*, *prejazdovaDlžka*, *Spravca_Skratka*, *Vlastnik_Skratka*, *dĺžka vozovky (m)*, *šírka vozovky*, *typKrytuVozovky*.

detailedRoad a **critical** dataframey som uložil na disk s encoding UTF-8.



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

3.3. Logstash

Vybral som si súbory, ktoré majú v sebe uložené údaje z **critical** a **hole** dataframu. Týkajú sa mojej témy - monitorovanie poškodených ciest.

K týmto súborom som definoval stĺpce, ktoré neobsahujú diakritiku a pridal som ich do indexu **global**. Niektoré atribúty (napr. *UsekoveStaniceZaciatku*, *UciMax*) potrebovali konverziu v logstash na číselný typ - float alebo int. Rovnako som pomocou filtru mutata zmenil atribút *DatumZberu* na dátum a určil som ho ako *@timestamp*.

```
input {
  file {
    path => "/usr/share/logstash/data/secCriticalRating/*.csv"
    start_position => "beginning"
  }
  file {
    path => "/usr/share/logstash/data/ptp_linescanparovane-2019.csv"
    start_position => "beginning"
  }
}

filter{
  if [path] == "/usr/share/logstash/data/ptp_linescanparovane-2019.csv" {
    csv {
      skip_header => true
      separator => ";"
      columns => ["TriedaCK", "cisloCK", "usekID", "pociatocnyUzol",
        "koncovyUzol", "UsekoveStaniceZaciatku", "UsekoveStanicenienieKonca",
        "Krok merania / m", "UciMax", "UciPriem", "Umiestnenie", "TypParovania",
        "DatumZberu", "KMStanicenienieZaciatku_km", "KMStanicenienieKonca_km",
        "KumulStanicenienieZaciatku_m", "KumulStanicenienieKonca_m", "typMerania",
        "TypUseku", "SmerUseku", "Poradie", "EvidencnaDlzskaUseku_m",
        "Spravca_Skratka", "Vlastnik_Skratka", "Okres", "Kraj"]
    }
    mutate {
      convert => {
        "usekID" => "integer"
        "cisloCK" => "integer"
        "UsekoveStaniceZaciatku" => "integer"
        "UsekoveStanicenienieKonca" => "integer"
        "Krok merania / m" => "float"
        "UciMax" => "float"
        "UciPriem" => "float"
        "KMStanicenienieZaciatku_km" => "float"
        "KMStanicenienieKonca_km" => "float"
        "KumulStanicenienieZaciatku_m" => "float"
        "KumulStanicenienieKonca_m" => "float"
        "Poradie" => "integer"
      }
    }
    date {
      match => [ "DatumZberu", "d.M.yyyy" ]
      target => "@timestamp"
    }
  }
}
```

Výstrižok z konfiguračného súboru



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

3.4. Elastic Search

3.4.1. Query

Vypíš všetky záznamy, ktoré majú usekID == 344. Potrebujeme vidieť detail jedného úseku CK.

```
GET _search
{
  "query": {
    "match": {
      "usekID": "344"
    }
  }
}
```

Vypíš všetky úseky CK v Nitrianskom kraji, zoradené podľa čísla cesty. Úseky patriace do ciest s nižším číslom sú vždy dôležitejšie CK (cesta 1 predstavuje prvú diaľnicu, cesta 9999 bude miestna cesta s nízkym významom)

```
GET _search
{
  "sort": {
    "cisloCk": {
      "order" : "desc"}
  },
  "query": {
    "match": {
      "Kraj": "Nitriansky kraj"
    }
  }
}
```

Zobraz všetky úseky, ktoré končia v uzle 3424XXXXXX - križovatka v Skalici. Nájďme všetky úseky, ktoré smerujú k tejto križovatke.

```
GET _search
{
  "query": {
    "wildcard": {
      "koncovyUzol": "3424*"
    }
  }
}
```

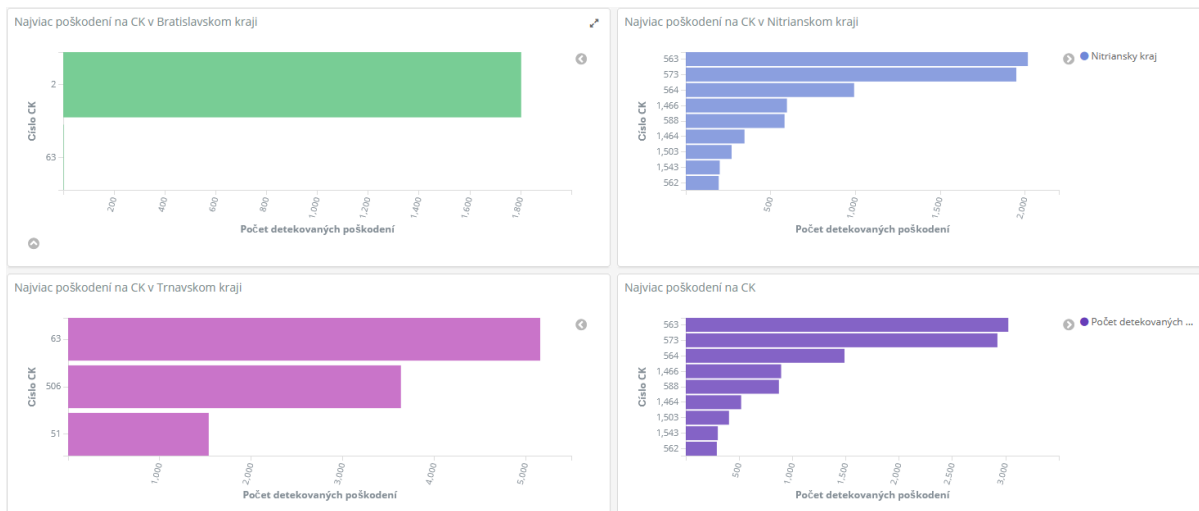


3.4.2. Vizualizácie

Dashboard 1: Overview na stav poškodení ciest



Dashboard 2: Overview na poškodenie podľa krajov Západného Slovenska



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

4. Záver

4.1. Zhodnotenie

Z môjho pohľadu bola práca na predmete BI-BIG *veľkým prínosom* a zaujímavou prvou skúsenosťou s novými technológiami, ktoré stoja hlavne na **distribuvaní** výpočtu/úložiska. V tejto práci som sa naučil pracovať so základmi technológie ELK stacku ale bohužiaľ som vôbec nevyužil možnosti nástrojov - *učil som sa základy*, ktoré som cez rok nezískal.

Dlhší čas som sa snažil pochopiť a použiť riešenia, ktoré sme preberali na hodine (ukážkové riešenie práce s nehodami v NYC) ale nakoniec som si všetko musel zistiť sám. Podobne to bolo aj s ostatnými komponentami ELK stacku. Nedostatočná dokumentácia rozdelená do rôznych verzií mi zabrala dosť času, nebol som zvyknutý na takéto neprijemnosti ale teraz *mám o skúsenosť viac*. Účastnil som sa všetkých cvičení. Pre nedostatok času som nemal možnosť vytvoriť zaujímavejšie agregácie a rozumnejšie využiť údaje - spájanie, vyťaženie znalostí atď.

4.2. Zdroje

<https://courses.fit.cvut.cz/BI-BIG>
<https://www.elastic.co/what-is/elk-stack>
<https://www.ssc.sk/sk/>
<https://www.kaggle.com/>
<https://logz.io/learn/complete-guide-elk-stack/>
<https://gitlab.fit.cvut.cz/sutymate/bi-big-2020>

