

# The State of JavaScript 2016

---

Seminarski rad u okviru kursa Istraživanje  
podataka

**Marija Filipović i Lazar Bojanić**

**9/18/2018**

# Sadržaj

<b>UVOD.....</b>	<b>3</b>
UPOZNAVANJE SA PODACIMA .....	3
<b>PREDSTAVLJANJE PODATAKA .....</b>	<b>4</b>
PRETPROCESIRANJE I ANALIZA PODATAKA .....	4
<i>Obrada nedostajućih vrednosti i elementi van granice.....</i>	<i>6</i>
VIZUELIZACIJA PODATAKA.....	8
<b>PRAVILA PRIDRUŽIVANJA .....</b>	<b>11</b>
<b>KLASTEROVANJE .....</b>	<b>15</b>
<b>KLASIFIKACIJA .....</b>	<b>21</b>

## Uvod

### Upoznavanje sa podacima

Naziv baze podataka nad čijim podacima radimo u ovom projektu je “The state of JavaScript”, i ona sadrži odgovore na anketu sprovedenu 2016. godine od strane kompanije State of JS nad preko devet hiljada programera. Anketa se sastoji od raznovrsnih pitanja, počev od izbora front-end biblioteka i alata, do pitanja u kojima je težnja da se sazna u kojoj meri se uopšte JavaScript tehnologije susreću sa različitim potrebama i željama ispitanika. Informacije o bazi podataka, kao i sama baza nalaze se na linku: <https://www.kaggle.com/integis/state-of-javascript-2016>.

Atributi baze, čiji broj prelazi 90, predstavljaju pitanja postavljena velikom broju ispitanika, koji variraju po radnom iskustvu, položaju u firmi, veličini firme u kojoj su zaposleni, plati na godišnjem nivou, i mnogim drugim karakteristikama. Odgovori su dati kao vrednosti atributa numeričkog ili kategoričkog tipa, pri čemu su vrednosti numeričkih atributa isključivo celobrojne, dok vrednosti kategoričkih atributa predstavljaju odgovore na pitanja o poznavanju određene JavaScript tehnologije, i mogu imati jedan od narednih oblika:

- Never heard of it
- I’ve heard of it, and would like to learn it
- I’ve heard of it, and am not interested
- I’ve used it before, and would use it again
- I’ve used it before and would not use it again

Dakle, u pitanju koje kao odgovor očekuje atribut kategoričkog tipa nekog od navedenih oblika, sadržan je naziv neke JavaScript tehnologije koja je jedan od predstavnika određene grupe tehnologija, na primer:

- React, Angular, Angular 2, Ember, Backbone... pripadaju grupi Front-End biblioteka
- Redx, MobX, Relay...pripadaju grupi State Management biblioteka
- Mocha, Jasmine, Cucumber, Ava...pripadaju grupi Testing frameworks

Vrednosti numeričkih atributa predstavljaju ocene ispitanika dodeljene određenoj JavaScript tehnologiji, i uzimaju vrednosti od 0 do 5. Neka od ovakvih pitanja su na primer:

- On a scale of one to five cats, how happy are you with your current solution for the front-end?
- On a scale of one to five thunderbolts how happy are you with your current solution for the State Management?
- On a scale of one to five trophies how happy are you with your current full-overflow solution?

Krajnji numerički atributi predstavljaju pitanja koja kao odgovore takođe imaju celobrojne vrednosti od 0 do 5, međutim, ovi odgovori su mera zadovoljnosti ispitanika samim JavaScriptom kao glavnim programskim jezikom korišćenja, njegovim brzim razvojem, njegovom kompleksnošću, itd.

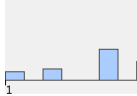
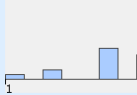
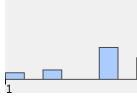
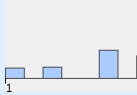
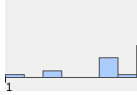
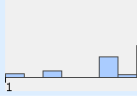
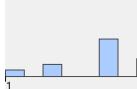
Krajnji kategorički atributi su pitanja o ličnim informacijama o ispitaniku koje bi mogle biti korisne za istraživanje podataka, poput broja godina njegovog radnog iskustva, veličini kompanije u kojoj je zaposlen, iznosu njegove plate na godišnjem nivou, pa čak i informacija o njegovom omiljenom tekstualnom editoru.

Projekat je rađen koristeći KNIME Analytics alat i IBM SPSS Modeler.

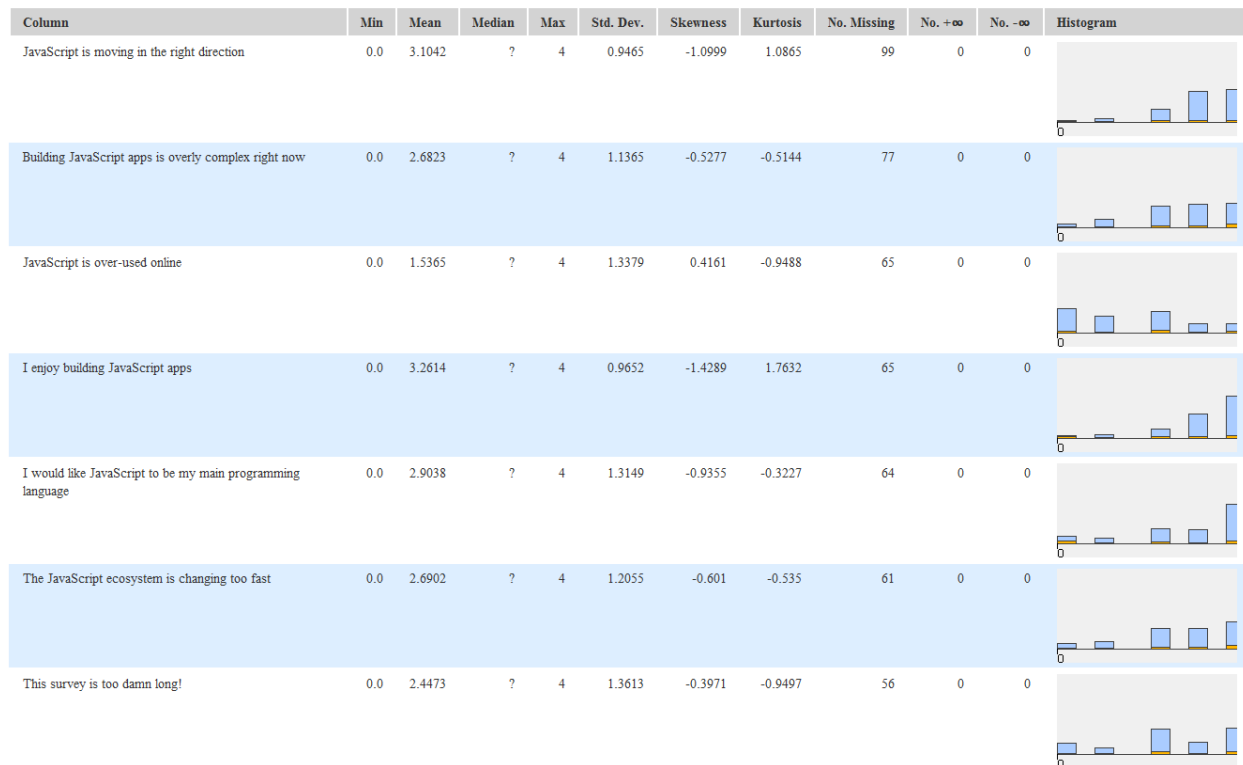
## **Predstavljanje podataka**

### **Pretprocesiranje i analiza podataka**

Za predstavljanje podataka koristimo KNIME. Podatke čitamo pomoću čvora Excel Reader. Na slikama 1 i 2 se mogu videti neki od numeričkih atributa i njihove osnovne statističke osobine poput min, max, median, mean, st.dev, itd. dobijene čvorom Statistics.

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
On a scale of one to five dogs, how happy are you with your current flavor of JavaScript?	1	3.9502	?	5	0.8479	-0.8323	1.049	0	0	0	
On a scale of one to five cats, how happy are you with your current solution for the front-end?	1	3.7777	?	5	0.9212	-0.585	0.2091	0	0	0	
On a scale of one to five thunderbolts, how happy are you with your current solution for state management?	1	3.1854	?	5	1.1424	-0.2153	-0.537	0	0	0	
On a scale of one to five crowns, how happy are you with your current solution for the API layer?	1	3.3995	?	5	1.0416	-0.369	-0.1722	0	0	0	
On a scale of one to five trophies, how happy are you with your current full-stack solution?	1	3.2916	?	5	1.0884	-0.3564	-0.2719	0	0	0	
On a scale of one to five severed hands, how happy are you with the current state of JavaScript testing?	1	3.1729	?	5	1.1705	-0.3151	-0.619	0	0	0	
On a scale of one to five lightbulbs, how happy are you with the current state of CSS?	1	3.7487	?	5	0.9856	-0.6601	0.1373	0	0	0	
On a scale of one to five droplets, how happy are you with the current state of build tools?	1	3.6453	?	5	1.0314	-0.6832	0.145	0	0	0	
On a scale of one to five pencils, how happy are you with the current state of mobile apps?	1	3.111	?	5	1.0166	-0.1515	-0.1364	0	0	0	

**Slika 1 - Statistika numeričkih atributa koji uzimaju celobrojne vrednosti i predstavljaju ocene JavaScript tehnologija**



**Slika 2 – Statistika numeričkih atributa koji uzimaju celobrojne vrednosti i predstavljaju mere zadovoljnosti ispitanika određenim osobinama JavaScript tehnologija**

## Obrada nedostajućih vrednosti i elementi van granice

Proveravamo da li imamo nedostajuće vrednosti koristeći čvor Missing value. Nakon postavke da se pri pojavi nedostajuće vrednosti red koji je sadrži obriše, primećujemo da je izlazna tabela sa takvom obradom nedostajućih vrednosti ostala bez ijednog reda, što nam nije cilj. Numerički atributi su celobrojne vrednosti, i želimo da takvi i ostanu, stoga u Missing Value čvoru za obradu nedostajućih vrednosti u slučaju pojave u koloni numeričkih atributa biramo opciju Fixed Value, i postavljamo je na 3. Na ovaj način, ako polje nije bilo popunjeno u bazi, nakon izlaska iz Missing Value čvora ono će biti postavljeno na 3, što nam odgovara jer na ovaj način polja numeričkog tipa koja nisu popunjena neće dovesti do problema prilikom obrada podataka, a takođe neće ni menjati rezultate istraživanja jer je 3 neutralna vrednost u našem slučaju. Za kategoričke attribute postavljamo u Missing value čvoru opciju Fixed value na "NA" sto je skraćenica za "Not available", i isto kao i kod numeričkih atributa i ovde označava da polje ankete nije bilo popunjeno.

Kako bismo uočili elemente van granica u bazi, koristimo čvor Box Plot, koji nam omogućava da u Box Plot View uočimo elemente van granica i obeležimo ih – naglasimo ih postavljanjem HiLite selected

The diagram illustrates a data processing workflow. It starts with 'Excel Reader (XLS)' (Node 1), which feeds into four parallel components: 'Missing Value' (Node 2), 'Box Plot' (Node 3), 'Statistics' (Node 5), and 'Interactive Table' (Node 4). Each component is represented by a distinct icon and a node label.

```

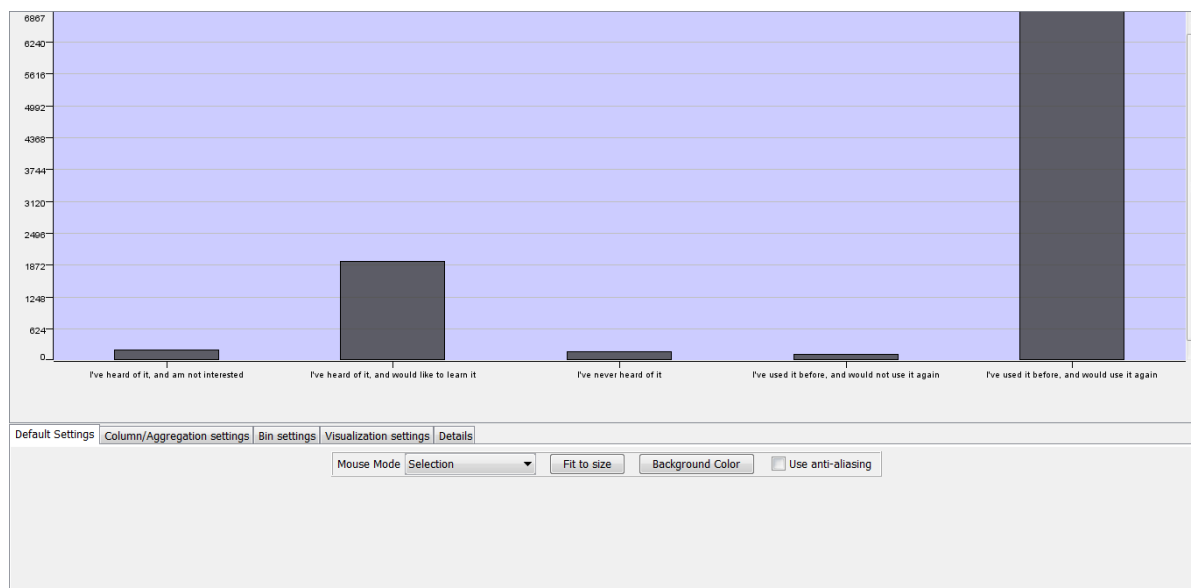
graph LR
    Node1[Excel Reader XLS Node 1] --> Node2[Missing Value Node 2]
    Node1 --> Node3[Box Plot Node 3]
    Node1 --> Node4[Interactive Table Node 4]
    Node1 --> Node5[Statistics Node 5]
  
```

7

## Vizuelizacija podataka

Kategorički atributi baze uglavnom predstavljaju odgovore ispitanika koji označavaju ocenu određene JavaScript tehnologije u njihovom iskustvu do sada. Ovakvih atributa ima mnogo u bazi i mogu se odnositi na različite oblasti primene JavaScripta, pa smo odlučili da predstavimo samo podatke iz jedne grupe – JavaScript Flavors.

U grupi JavaScript Flavors postoji nekoliko navedenih tehnologija, od kojih svaka predstavlja po kolonu u bazi: ES6, Plain JavaScript, TypeScript, Elm, ClojureScript, itd. Prilikom istraživanja koliko ispitanici poznaju ove tehnologije i koliko su njima zadovoljni, primetili smo da se dve tehnologije – ES6 i Elm značajno razlikuju po učestalosti jednog odgovora: “I’ve never heard of it”. Slike 5 i 6 pomoću histograma pokazuju drastično različit broj pojavljivanja ovog odgovora u slučaju biblioteka ES6 i Elm. Željene kolone ES6 i Elm smo pojedinačno izdvojili pomoću čvora Column Filter, a zatim podatke vizuelno predstavili histogramima pomoću čvora Histogram (interactive).

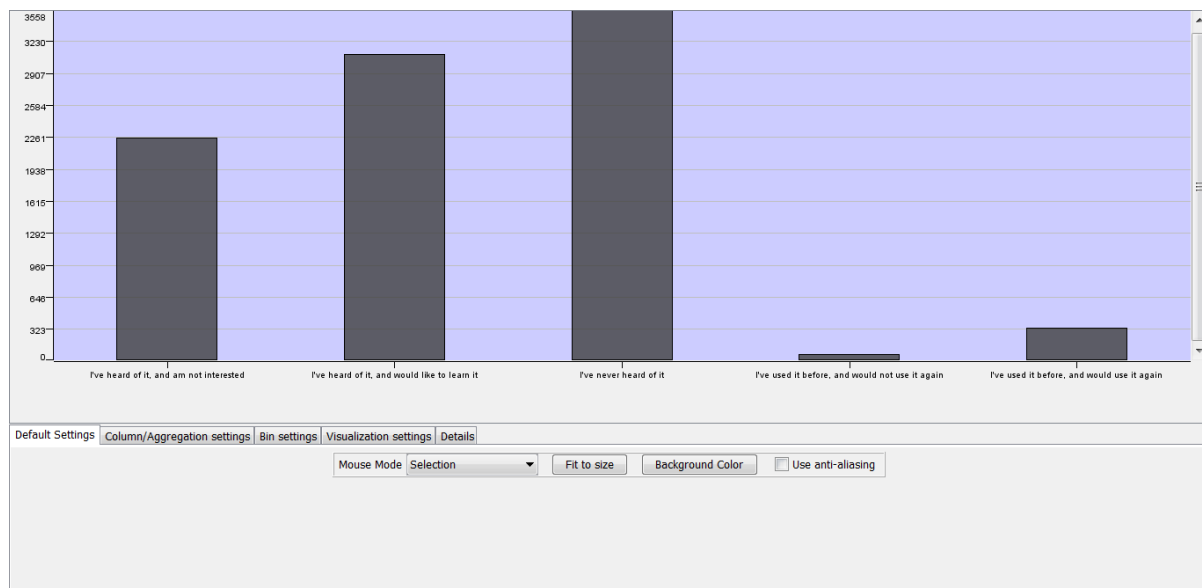


**Slika 5 – Možemo primetiti nizak broj pojavljivanja odgovora “I’ve never heard of it” u slučaju biblioteke ES6**

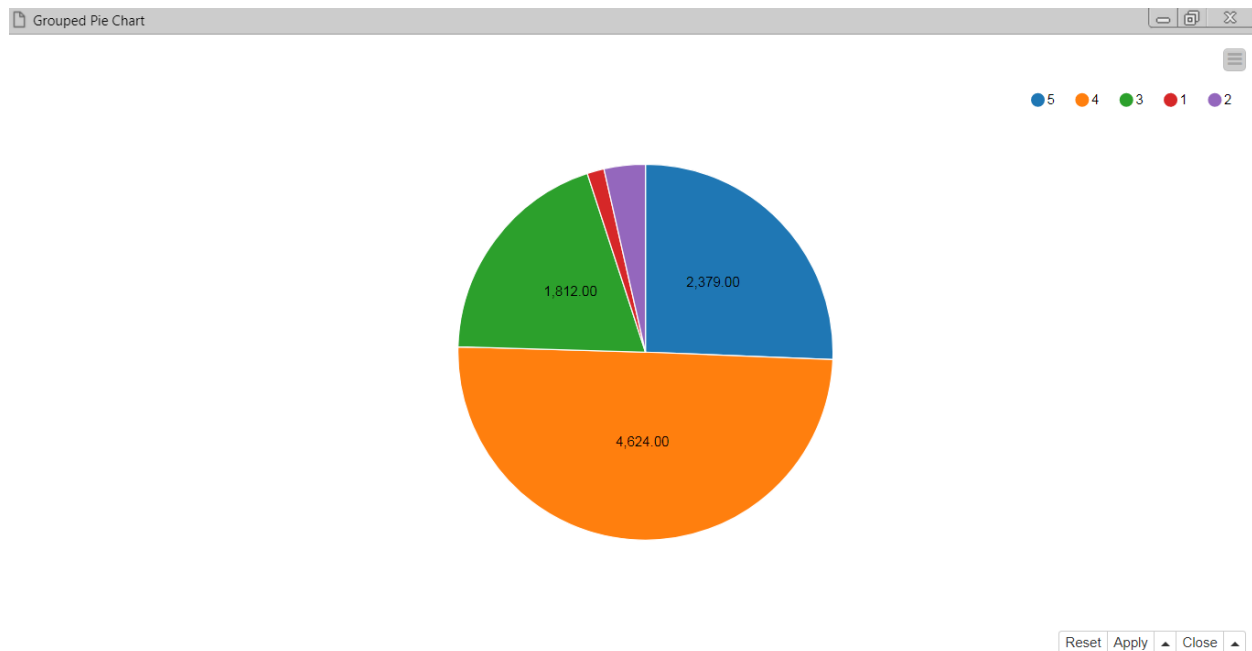
Sa slika jasno zaključujemo da se ovaj odgovor jako retko javljao u slučaju biblioteke ES6 – manje od 600 puta, dok je ovaj odgovor naveden od strane preko čak 3000 ispitanika kad je reč o biblioteci Elm.



Numerički atributi uglavnom predstavljaju celobrojne vrednosti kao ocene određene tehnologije. Izabrali smo da vizuelno prikažemo učestalosti pojavljivanja odgovora na pitanje “On a scale of one to five dogs, how happy are you with your current JavaScript flavor”. Koristili smo Column Filter čvor da izdvojimo kolonu s ovim pitanjem, zatim izlaz iz ovog čvora naveli kao ulaz čvora Number To String kako bismo imali odgovarajući ulaz za čvor Pie/Donut Chart (JavaScript) kojim završavamo prikaz ovih podataka. Na slici 7 se može uočiti da su korisnici prilično zadovoljni u ovom segmentu, sa ocenom 4 kao najučestalijim odgovorom.

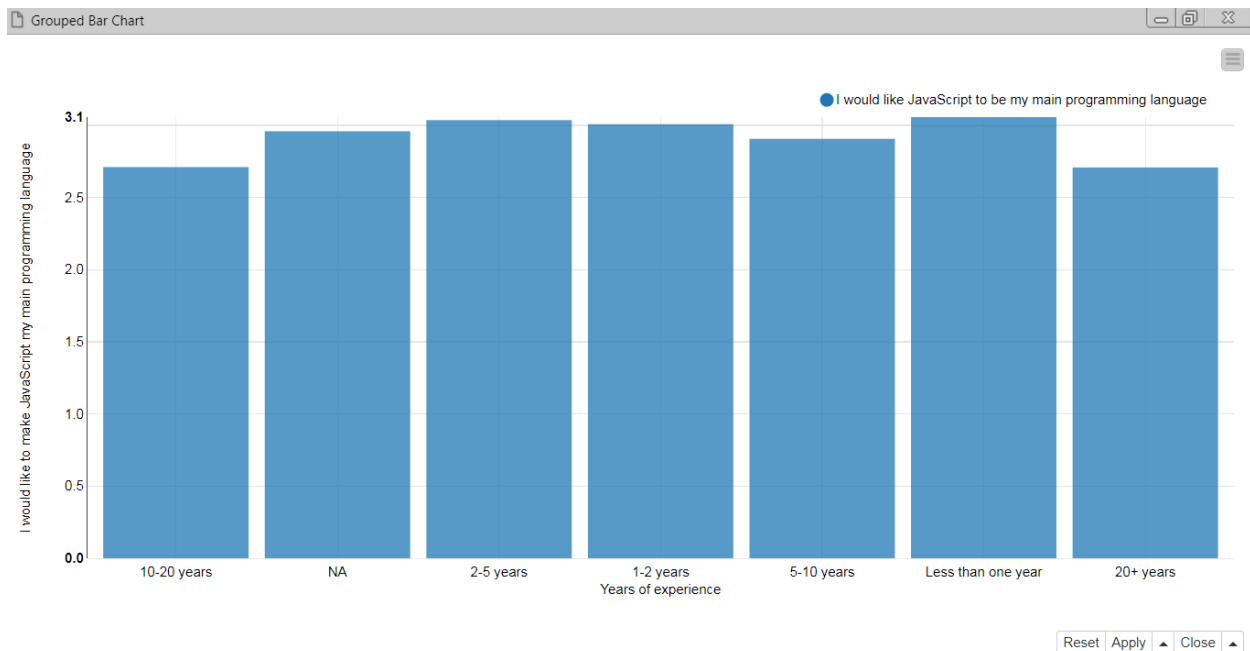


**Slika 6 - Možemo primetiti visok broj pojavljivanja odgovora “I’ve never heard of it” u slučaju biblioteke Elm**



**Slika 7 – Broj pojavljivanja svake ocene od 1 do 5 kao odgovor ispitanika na pitanje "On a scale of one to five dogs, how happy are you with you current JavaScript flavor"**

Pored ovih atributa, takođe imamo i attribute koji označavaju odgovore na pitanja koja se odnose na same ispitanike i kao takva predstavljaju zanimljivija pitanja i ona kod kojih možemo uočiti najviše zanimljivih informacija. Pri vizuelizaciji ovih podataka odlučili smo da predstavimo odnos između broja godina iskustva programera i njegove želje da radi u JavaScriptu kao glavnom programskom jeziku. Izabrali smo da prikazemo prosečne odgovore ispitanika pomoću čvora Bar Chart (JavaScript).



**Slika 8 – Prikazani prosečni odgovori ispitanika različitog broja godina radnog iskustva na pitanje da li bi želeli JavaScript kao svoj glavni programski jezik**

## Pravila pridruživanja

Zahvaljujući pravilima pridruživanja imamo mogućnost da predvidimo pojavljivanja neke stavke na osnovu pojavljivanja drugih stavki u transakcijama. Za dati skup transakcija  $T$  cilj određivanja pravila pridruživanja je pronaći sva pravila koja imaju:

- podršku (support)  $\geq$  minsup (minimalni prag podrške)
- pouzdanost (confidence)  $\geq$  minconf (minimalni prag pouzdanosti)

Definicija pravila pridruživanja:

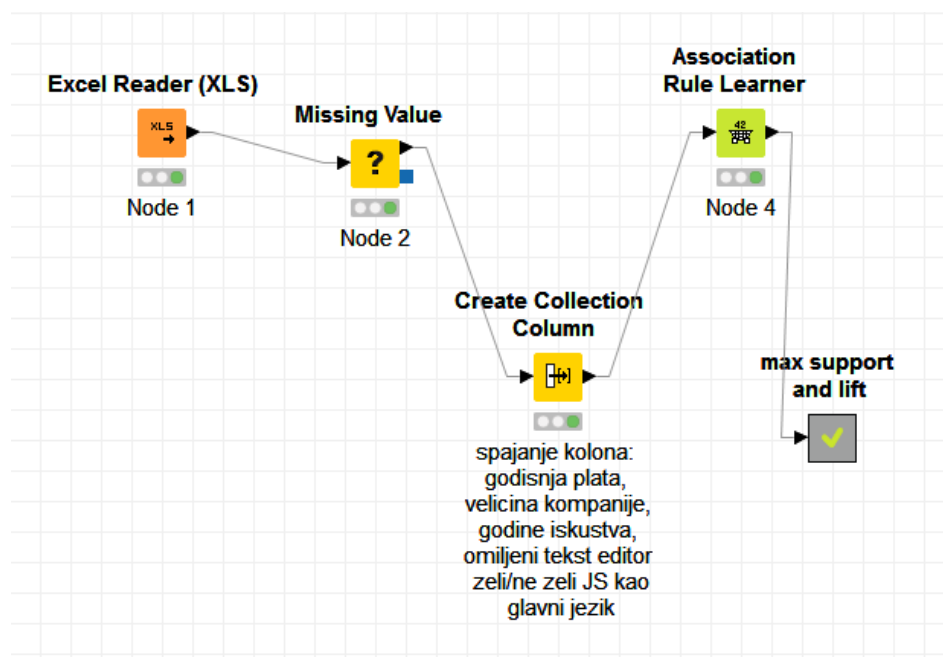
Neka su  $X$  i  $Y$  dva skupa stavki. Tada se pravilo u oznaci  $X \Rightarrow Y$  naziva pravilo pridruživanja sa minimalnom podrškom minsup i minimalnom pouzdanošću minconf ako važi:

- Podrška stavke  $X \cup Y$  je veća ili jednaka od minsup
- Pouzdanost pravila  $X \Rightarrow Y$  je veća ili jednaka od minconf

Najpre pročitamo podatke i obradimo nedostajuće vrednosti na način koji smo opisali. Zatim čvorom Create Collection Column pravimo listu koja obuhvata vrednosti iz sledećih kolona:

- I would like JavaScript to be my main programming language
- Years of experience
- Company Size
- Yearly Salary
- Favorite Text Editor

Sad imamo odgovarajući ulaz za čvor Association Rule Learner koji zahteva ulaz kolekcijskog tipa, odnosno transakciju. Za velike baze podataka je uobičajeno da se minimal support postavi na vrednost između 0.05 i 0.1, međutim, u slučaju naše baze zanimljiviji rezultati se dobiju postavljanjem ove vrednosti na 0.01. Vrednost za minimal confidence postavljamo na 0.6.



**Slika 9 – Čvorovi korišćeni pri dobijanju pravila pridruživanja**

Kada pogledamo dobijena pravila pridruživanja uočavamo da u koloni Lift pored vrednosti koje se kreću između 1 i 2 takođe postoje i vrednosti koje su znatno veće – između 5 i 6. Ovakva pravila, koja uzimaju vrednosti za lift preko ili ispod 1 smatramo zanimljivim, dok pravila koja imaju lift vrednosti jednake sa 1 smatramo nezanimljivim jer su kod njih dobijeni rezultati jednaki i očekivanim. Međutim, primećujemo da, iako bi pravila pridruživanja sa ovako visokim lift vrednostima trebalo da daju neočekivane rezultate, ipak u njima možemo zapaziti logičnosti. Među ovakvim redovima imamo na primer red:

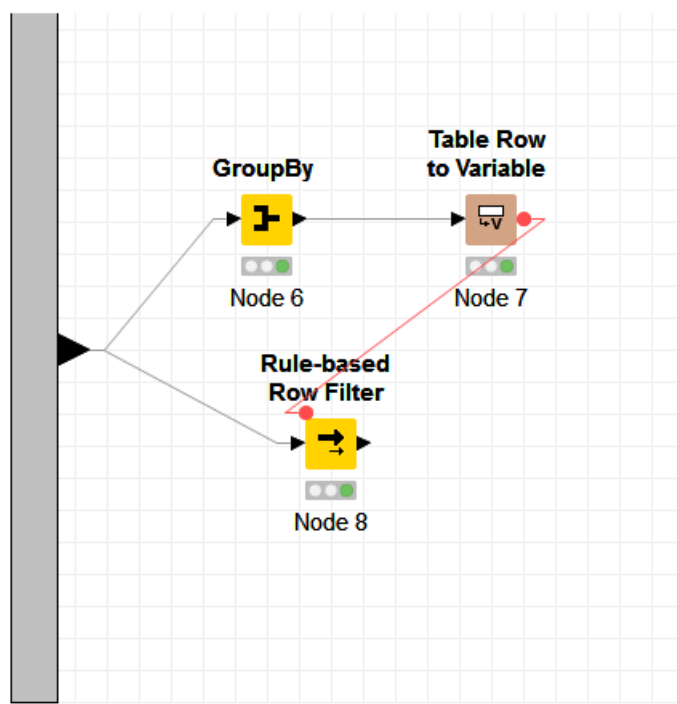
Just me ← [1-2 years, \$30 – 50k, Atom]

Što znači da je verovatno da ispitanik sa radnim iskustvom od 1 do 2 godine, platom između 30 i 50 hiljada dolara godišnje, koji koristi Atom, samostalno radi, na osnovu čega bismo mogli zaključiti da je freelancer. Nasuprot ovome, postoji i red sa:

Just me ← [2-5 years, I work for free ☺]

sa vrednosti za lift sa 5.721, ali za razliku od prethodnog primera, sada je polje kolone Yearly Salary I work for free. Na osnovu ovog primera možemo zaključiti da, iako pravilo uzima lift vrednost tako da bi trebalo da bude zanimljivo, ono ipak nije savršeno, jer nam je kao neočekivan rezultat dao sasvim logičan odgovor - da ispitanik koji nema zaradu verovatno radi samostalno, odnosno nije zaposlen u kompaniji.

Kako bismo izdvojili pravila sa najvišim lift i support vrednostima, koristimo čvor Group By u kojem za lift i support kolone postavljamo Aggregation na Maximum, i izlaz iz tog čvora koristimo kao ulaz za čvor Table Row to Variable, a zatim njegov izlaz unosimo u čvor Rule Based Row Filter. U ovom čvoru postavljamo izraz kao na slici 11, čime izdvajamo samo redove sa maksimalnim vrednostima za lift ili support.



Slika 10 – Prikaz čvorova kojim izdvajamo samo redove sa maksimalnom vrednošću za lift i support

Expression

```
1 // enter ordered set of rules, e.g.:
2 // $double column name$ > 5.0 => FALSE
3 // $string column name$ LIKE "*blue*" => FALSE
4 // TRUE => TRUE
5 $$Support$ = $$DMax*(Support)}$$ => TRUE
6 $Lift$ = $$DMax*(Lift)}$$ => TRUE
```

Slika 11 – Izraz u Rule Based Row Filter čvoru za uparivanje redova sa max lift i support vrednostima

D Max*(Lift)	D Max*(Support)
5.931	0.048

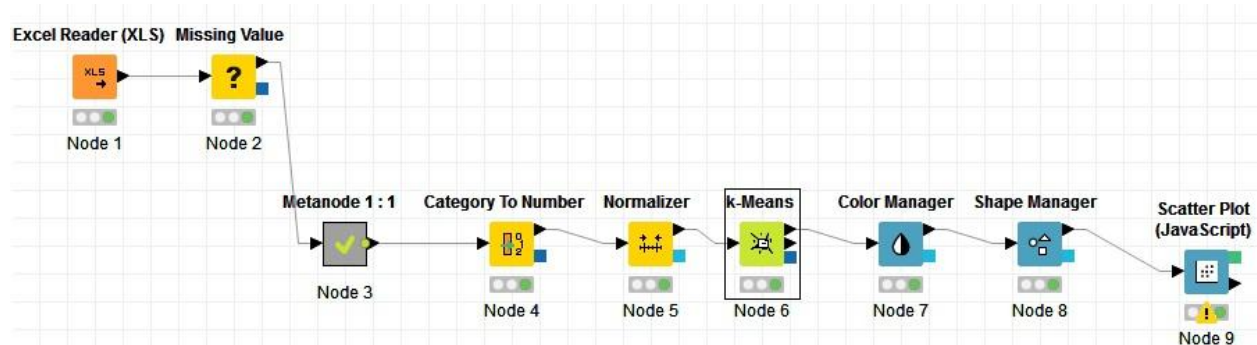
Slika 12 – Maksimalne vrednosti za Lift i Support dobijene pomoću Group By čvora

## Klasterovanje

Klasterovanje je način pronalaženja grupa objekata takvih da su objekti u grupama međusobno slični, dok su objekti u različitim grupama, tj. klasterima, međusobno različiti.

U ovom delu, klasterovanjem ćemo pokušati da utvrdimo da li postoji veza između broja framework-a koje poznaje programer i njegove godišnje plate, a nakon toga ćemo pokušati da utvrdimo da li postoji veza između odgovora na pitanje vezano za ES6 i odgovora na pitanje vezano za Typescript. Prvo istraživanje ćemo obraditi korišćenjem alata Knime, dok ćemo drugo obraditi korišćenjem alata SPSS Modeler.

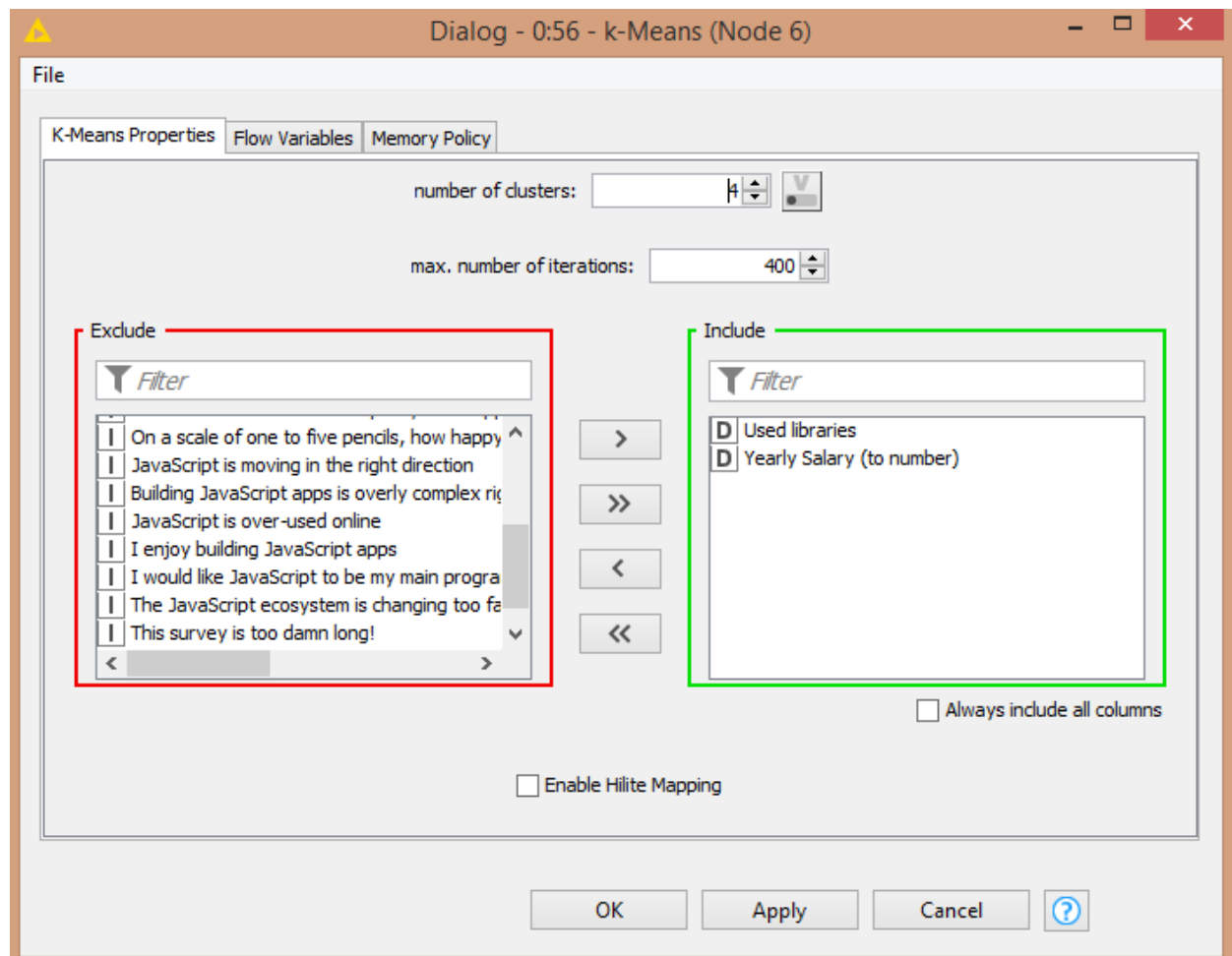
Na skupu podataka StateofJS primenićemo algoritam k-sredina koji na početku obeleži nasumičnih k instanci skupa kao centroide klastera, pridružuje instance najbližim centroidima koristeći euklidsko rastojanje, preračunava centroide klastera i ponavlja postupak sve dok se klasteri menjaju ili dok se ne prekorači određeni broj iteracija.



Slika 13 - Workflow u Knime alatu kojim se vrši klasterovanje algoritmom k-Means

Nakon učitavanja podataka i njihove obrade, dodat je atribut `used libraries` koji za svaki red računa koliko puta je upisan odgovor „I’ve used it before, and would use it again“ i na taj način je određeno koliko programer zaista poznaje oblast za koju je postavio ovakav odgovor i pretpostavićemo da ovaj odgovor znači da programer dobro vlada određenom tehnikom.

Nakon prebacivanja kategoričkog atributa “Yearly salary” u numerički, atributi “Yearly salary” i “Used libraries” su normalizovani i prosleđeni čvoru k-means koji će izvršiti klasterovanje prema zadatom parametru k i prema zadatim atributima.



**Slika 14 - Konfigurisanje čvora k-Means**

Sa brojem čvorova i brojem iteracija je eksperimentisano i dobijeni su različiti rezultati. Nakon izvršavanja algoritma k-sredina klasteri su predstavljeni različitim oblicima i bojama i vizuelizovani koristeći čvor “Scatter plot”.



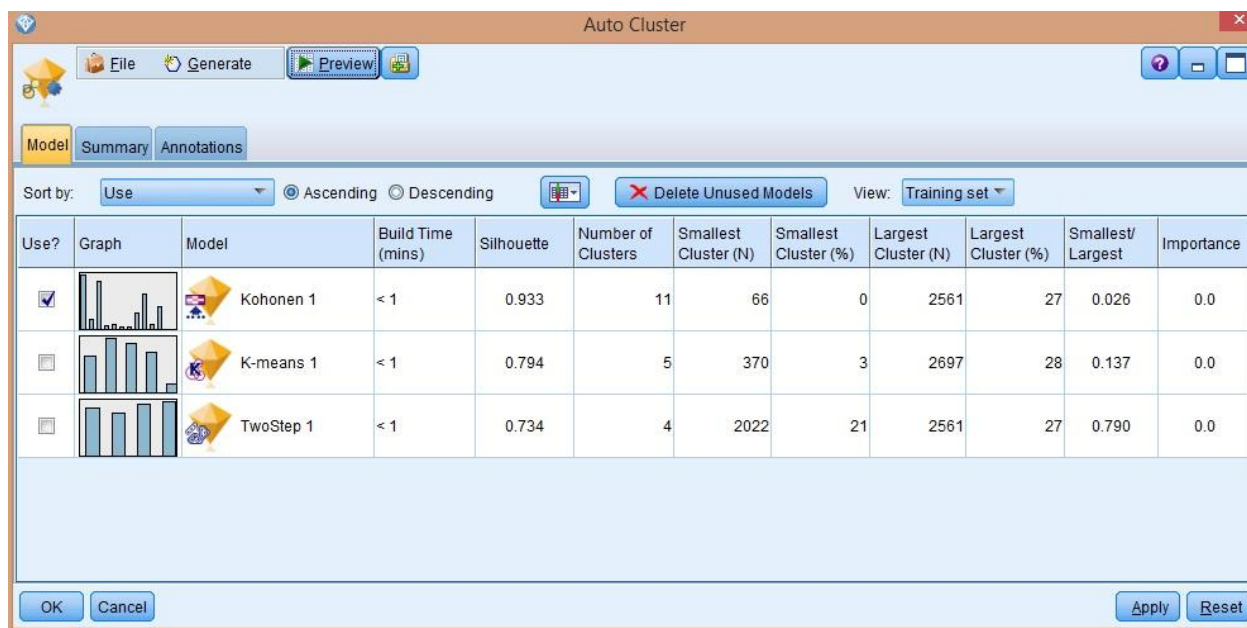


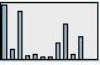


Slika 15 - Predstavljanje rezultata klasterovanja algoritmom k-Means koristeći različit k parametar

Koji god bio parametar k u algoritmu k-sredina, dobijamo dobro razdvojene klastere čime potvrđujemo da su atributi „Used libraries“ i „Yearly salary“ povezani, što je i logično jer što više jezika i biblioteka programer poznaje to mu više raste znanje, potražnja a samim tim i godišnja plata.

Sledeće želimo u alatu SPSS Modeler da ustanovimo da li postoji veza izmedju odgovora na pitanje vezano za ES6 i odgovora na pitanje vezano za Typescript. Ovom analizom želimo da saznamo da li programeri koji poznaju ES6 žele da nauče Typescript ili su ga već naučili, ili žele da se pridržavaju ES6 i da nastave da se dodatno usavršavaju u njemu.

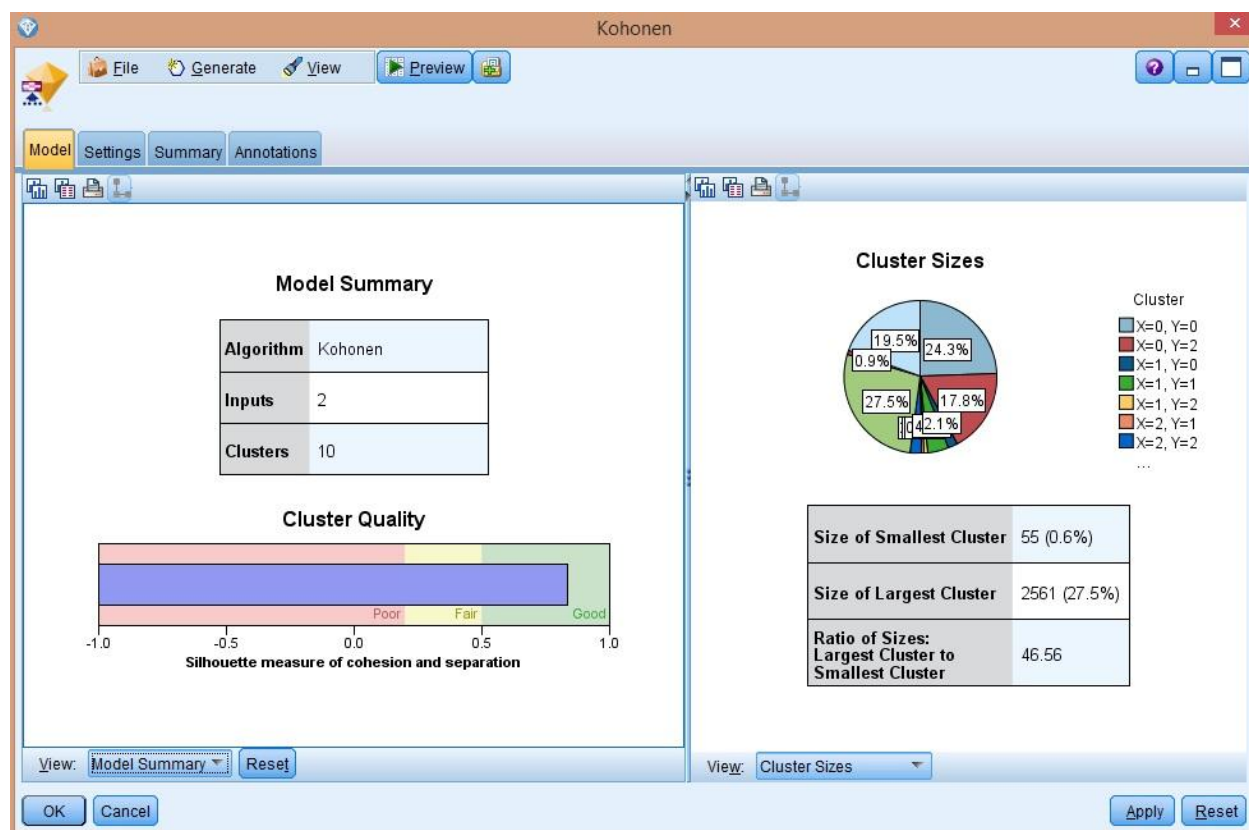
Koristićemo čvor „Auto cluster“ i proslediti mu attribute „ES6“ i „Typescript“.



Use?	Graph	Model	Build Time (mins)	Silhouette	Number of Clusters	Smallest Cluster (N)	Smallest Cluster (%)	Largest Cluster (N)	Largest Cluster (%)	Smallest/Largest	Importance
<input checked="" type="checkbox"/>		Kohonen 1	< 1	0.933	11	66	0	2561	27	0.026	0.0
<input type="checkbox"/>		K-means 1	< 1	0.794	5	370	3	2697	28	0.137	0.0
<input type="checkbox"/>		TwoStep 1	< 1	0.734	4	2022	21	2561	27	0.790	0.0

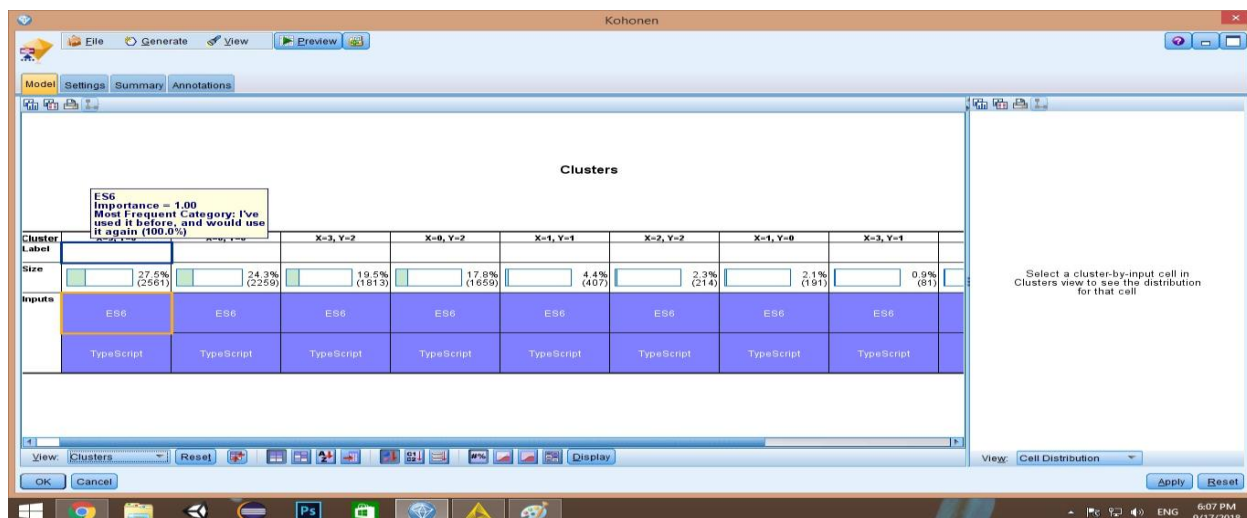
**Slika 16 - Rezultat čvora Auto Cluster**

Čvor je napravio 3 modela – koristeći algoritme Kohonen, k-sredina i TwoStep i sortirao modele prema silueti. Iz datog prikaza možemo primetiti da je najbolji rezultat dobijen koristeći algoritam Kohonen pa ćemo u nastavku analizirati rad ovog algoritma.



Slika 17 - Rezultat klasterovanja algoritmom Kohonen

U najveća tri klastera su ispitanici koji su na odgovore vezane za ES6 davali odgovore „I've used it before, and would use it again“, dok su za pitanje vezano za Typescript davali odgovore „I've heard of it, and would like to learn it“, „I've used it, and would use it again“ i „I've heard of it, and am not interested“.



Slika 18 - Vizuelizacija klastera

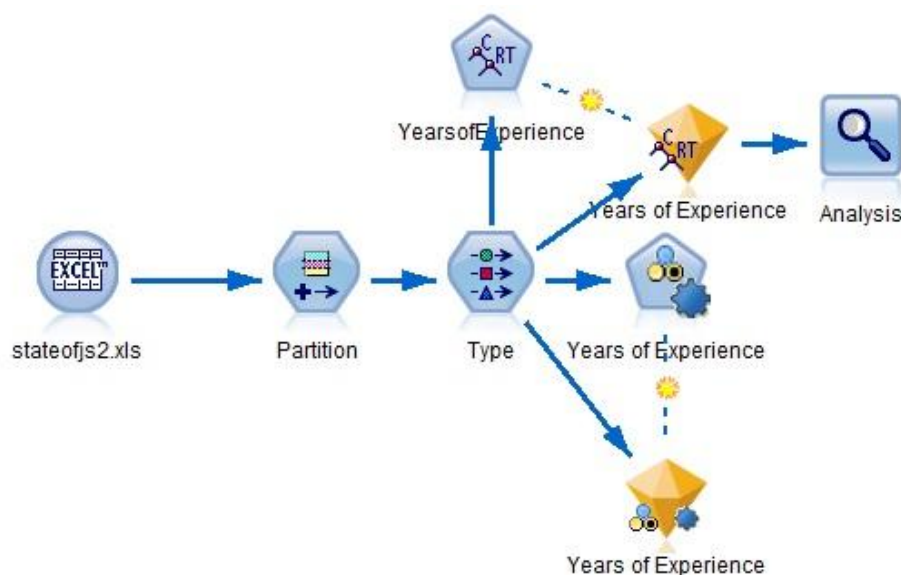
Zaključujemo da su ljudi najčešće davali odgovor da su koristili ES6 i da bi ga koristili u budućnosti dok su za Typescript davali uglavnom pozitivne odgovore – da su ili koristili i da bi nastavili u budućnosti ili da nisu probali ali da su zainteresovani. Ovakav rezultat je i logičan jer ES6 uglavnom koristi većina Javascript programera dok Typescript postaje jedan od popularnijih jezika sličnih Javascript-u pa otuda i interesovanje programera da ovladaju i ovom veštinom.

## Klasifikacija

U ovom poglavlju koristićemo klasifikaciju nad skupom podataka, tj. definisaćemo određene kategorije i dodeliti objekte skupa jednoj od njih. Pokušaćemo klasifikacijom da klasifikujemo podatke prema atributu „Years of experience“.

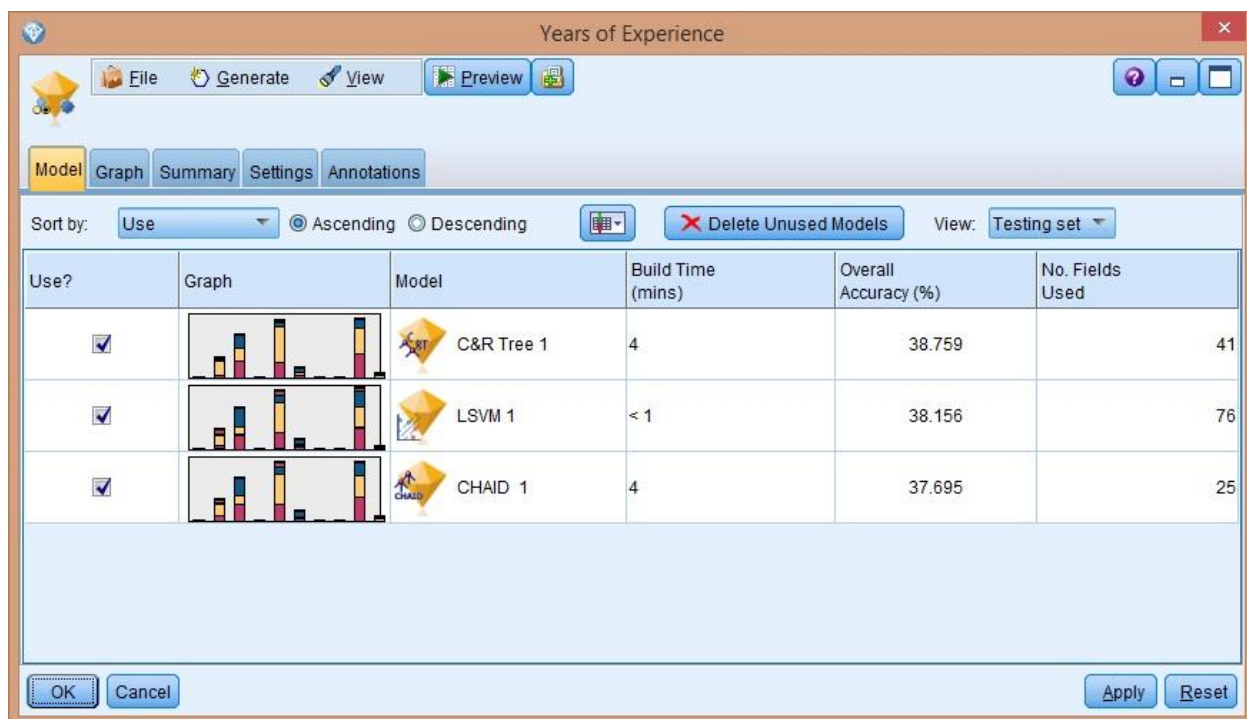
Particionisaćemo skup podataka na trening i test skup podataka, pri čemu trening skup sadrži 70% redova celog skupa, odnosno 6496 redova, a test skup sadrži ostalih 30% skupa, odnosno 2784 redova. Algoritmi će koristiti trening skup podataka da bi napravili model klasifikovanja, dok će na test skupu primeniti taj model.

Koristićemo alat SPSS Modeler za prikazivanje klasifikacije podataka.



**Slika 19 - Workflow u alatu SPSS Modeler koji vrši klasifikaciju podataka**

Prvo smo particionisali ulazne vrednosti na trening i test skup u zatom odnosu. Nakon toga smo prosledili podatke „Type“ čvoru u kome smo definisali šta je ciljna klasa, označili tipove podataka i označili koje podatke nećemo posmatrati u klasifikovanju. Podaci iz „Type“ čvora su prosledjeni „Auto classifier“ čvoru u kome smo označili da želimo da primenimo sve algoritme klasifikacije kako bismo iz rezultata odabrali najoptimalniji algoritam.



Slika 20 - Rezultat rada čvora Auto Classifier

Rezultat rada čvora “Auto classifier” su prikazani na slici i mogu se videti nekoliko najbolje rangiranih modela klasifikacije. Izabraćemo najbolje rangirani algoritam – C&R Tree algoritam i koristićemo Bajesovsku mrežu.

'Partition'	1_Training		2_Testing	
Correct	2,565	39.63%	1,093	38.57%
Wrong	3,907	60.37%	1,741	61.43%
Total	6,472		2,834	

Slika 21 - Rezultat klasifikacije algoritmom C&R Tree

'Partition'	1_Training		2_Testing	
Correct	2,585	39.94%	634	22.37%
Wrong	3,887	60.06%	2,200	77.63%
Total	6,472		2,834	

Slika 22 - Rezultat klasifikacije algoritmom Bajesovske mreže

C&R algoritam je na trening skupu uspešno klasifikovao 39.63% trening podataka i 38.57% test podataka što i nije dobro.

Bajesovska mreža je na trening skupu uspešno klasifikovala 39.94% trening podataka što je bolje nego kod C&R algoritma, dok je na test podacima uspešno klasifikovala samo 22.37% podataka što je loše.

Oba algoritma su se loše pokazala, pokušano je klasifikovanje i po drugim atributima ali su dobijani samo još lošiji rezultati. Pošto je skup podataka koji je obrađivan anketa, teško je klasifikovati podatke sa velikom preciznošću. Može se zaključiti da ne postoji neka posebna veza između godina iskustva i ostalih atributa u skupu podataka.