Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Igor Jončevski

# Napovedovanje lastništva podjetij na osnovi analize omrežij družbenikov

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Marko Bajec

Ljubljana, 2013

*Besedilo je oblikovano z urejevalnikom besedil LaTeX.*

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

*Tržaška 25*
*1001 Ljubljana, Slovenija*
*telefon: 01 476 84 11*
*01 476 83 87*
*faks: 01 426 46 47*
*01 476 87 11*
*www.fri.uni-lj.si*
*e-mail: dekanat@fri.uni-lj.si*

Št. naloge: 00143 / 2013
Datum: 3.9.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **IGOR JONČEVSKI**

Naslov: **NAPOVEDOVANJE LASTNIŠTVA PODJETIJ NA OSNOVI ANALIZE OMREŽIJ DRUŽBENIKOV**
**PREDICTING BUSINESS OWNERSHIP BASED ON BOARD COLLABORATION NETWORKS**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Predvidevanje sprememb v lastništvu podjetij ter pa razumevanje vzorcev povezovanja med družbeniki ima v današnjem času številne uporabe. Kandidat naj v diplomskem delu preuči sodobne pristope za napovedovanje sprememb v lastniški shemi podjetij. Pri tem naj podatke predstavi v obliki različnih omrežij družbenikov ter preizkusi izbrane metode in tehnike analize omrežij. Osredotoči naj se predvsem na mere središčnosti vozlišč ter algoritme analize povezav. Na podlagi pridobljenega znanja naj predlaga enostaven model za napovedovanje sprememb lastništva podjetij.

Mentor:

prof. dr. Marko Bajec

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:

prof. dr. Nikolaj Zimic

# Izjava o avtorstvu diplomskega dela

Spodaj podpisani Igor Jončevski, z vpisno številko **63070420**, sem avtor diplomskega dela z naslovom:

*Napovedovanje lastništva podjetij na osnovi analize omrežij družbenikov*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marka Bajca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 16. decembra 2013 Podpis avtorja:

*Na tem mestu bi se rad zahvalil družini, ki me je tekom celotnega študija spodbujala in mi vedno stala ob strani. Zahvala gre mentorju, izr. prof. dr. Marku Bajcu in as. dr. Lovru Šublju, za vso pomoč, koristne nasvete in smernice. Zahvalil bi se še doc. dr. Dejanu Lavbiču, ki je priskrbel podatke, na katerih temelji diplomsko delo. Izr. prof. dr. Leonu Oblaku bi se zahvalil za izjemno hitro popravljanje številnih slovničnih napak. Hvala tudi vsem ostalim, ki so na kakršenkoli način pomagali pri izdelavi tega diplomskega dela.*

На мојата фамилија.

# Kazalo

# Slike

# Tabele

# Povzetek

Namen diplomskega dela je bil izdelava napovednega modela, s katerim bi se lahko napovedovala sprememba lastništva v podjetju. Model je rezultat analize omrežja družbenikov, ki so zgrajena na podatkih o družbenikih in njihovi prisotnosti v podjetjih. Lastništvo je v bistvu stopnja vozlišča v omrežjih, ki so bila analizirana.

V procesu grajenja modela je bistvenega pomena obdelava podatkov in njihova predstavitev v podobi omrežja. To je bilo narejeno s pomočjo programskega jezika Java in programskih vmesnikov OWL in Gephi. Dobljena omrežja v formi grafov je bilo treba naprej analizirati. Nadaljnja analiza je poskrbela za pridobitev mere pomembnosti vozlišč v omrežju, ker je bila njihova uporaba ključnega pomena v procesu napovedovanja.

Dobljene mere so osnova za različne statistične metode, metode podatkovnega rudarjenja in metode strojnega učenja. Ugotovitve, pridobljene iz teh metod so pripeljale korak bližje do zamišljenega modela.

Končni rezultat je napovedni model, ki je lahko osnova za namizno ali spletno aplikacijo, ki bi lahko služila za napovedovanje, ne le spremembe lastništva, ampak tudi drugih podatkov.

**Ključne besede:** napovedni model, omrežje, graf, pomembnost vozlišč, statistika, podatkovno rudarjenje, obdelava podatkov, napovedovanje podatkov

# Abstract

The purpose of this thesis was the realization of a prediction model, with which we could predict the change of ownership in a network. The prediction model is a result of the analysis process of board collaboration networks, where the networks are in fact a representation of data for stockholders and their presence in a certain company. The ownership is represented with a node's degree in the networks which were analysed.

In the model realization process, data processing and their proper representation in the form of a network is essential. For that purpose, we used the Java programming language, coupled with the Application programming interfaces (APIs) OWL and Gephi. The resulting networks, represented as graphs, needed to be further analysed in order for us to acquire the node importance metrics of the network, which were crucial for the prediction process.

The acquired metrics are the basis for various statistical, data mining and machine learning methods. The results of those methods lead us to the creation of the model that we imagined in the first place.

The end result can be the basis for a desktop or a web application that could predict not just ownership change, but also other data.

**Keywords:** prediction model, network, graph, node importance, statistics, data mining, data processing, data prediction

# Poglavje 1

# Uvod

Vse je povezano: ljudje, informacije, mesta, dogodki. Odkar so se pojavili spletni družbeni mediji, se je povezanost med različnimi sferami življenja samo še stopnjevala. Praktičen način, da bi razumeli vse prepletene povezave, ki obstajajo je, da jih analiziramo kot družbena omrežja.

Glavna ideja diplomske naloge je napovedovanje spremembe lastništva podjetij, kar je pa dejansko predstavljena s spremembo stopnje vozlišč v omrežju. Za dosego tega cilja je bilo potrebno podatkovno množico, pridobljeno s pomočjo spletne ontologije (glej Poglavje 2.1.3) ustrezno obdelati, da bi iz nje dobili uporabne podatke. Slednje smo našli v poslovnih deležih, ki se nanašajo na družbenike in podjetja.

V Poglavju 2 smo se osredotočili na opisovanje vseh algoritmov, pristopov in metrik, ki so bili uporabljeni v celotnem procesu analiziranja omrežij, zgrajenih iz naših podatkov.

V Poglavju 3 smo se osredotočili na stvari, ki jih je bilo treba narediti, da bi se približali končnemu cilju, ki pa je pridobitev uporabnega napovednega modela. V Poglavju 3.1 smo se malo bolj poglobili v ontologijo, ki je bila uporabljena pri zajemanju uporabljenih podatkov. Nadaljevali smo s Poglavjem 3.2, kjer smo se spopadli s problemom analiziranja dvovrstnih omrežij in vprašanjem, zakaj analiza takšnih omrežij ni smiselna. Uporabljen je bil pristop projeciranja, ki eno dvovrstno omrežje spremeni v dveh enovrstnih.

Poglavje 3.3 opisuje pristope, ki smo jih uporabili pri analiziranju pridobljenih enovrstnih omrežij, kjer se kot rezultat pojavlja napovedni model v obliki klasifikacijskega drevesa.

Poglavje 4 vsebuje rezultate vseh metrik iz katerih se vidi, kako smo prišli do zgrajenega napovednega modela. Na preprostem primeru smo tudi razložili, kako se model uporablja.

V Poglavju 5 smo spet povzeli problem in pojasnili, kako smo prišli do končnega rezultata. Podali smo še nekoliko primerov, kako bi lahko analizo z uporabo različnih pristopov izboljšali in nadgradili.

# Poglavje 2

# Raziskovalna področja in uporabljene tehnologije

## 2.1 Spletne ontologije

### 2.1.1 Osnovni pojmi

Glavna naloga ontologije v računalništvu je predstavljanje podatkov in znanja, pridobljenega iz teh podatkov. Spletne ontologije se štejejo za enega od stebrov semantičnega spleta, čeprav za spletno ontologijo ne obstaja splošno sprejeta definicija. Na spletu obstaja veliko definicij o ontologijah v računalništvu, mnoge od njih pa so med seboj v nasprotju. V tej diplomski nalogi smo se osredotočili na definicijo[1], ki ontologijo opisuje kot formalni in izrečni opis konceptov v neki domeni, lastnosti vsakega koncepta, ki opisujejo različne značilnosti ali atribute določenega koncepta in omejitve lastnosti. Obstaja veliko razlogov, zakaj razviti ontologijo. Razlog, ki sem ga izbral jaz, je analiza baze znanja neke domene. Ontologija, skupaj s posameznimi primerki razredov predstavlja bazo znanja. Ontologija je sestavljena iz več komponent, in sicer:

- Posamezniki (angl. Individuals): primerki ali objekti (osnovni objekti).
- Razredi (angl. Classes): množice, zbirke, koncepti, vrste objektov ali

vrste stvari.

- Atributi (angl. Attributes): vidiki, lastnosti, značilnosti ali parametri, ki jih lahko objekti ali razredi imajo.
- Odnosi (angl. Relations): načini, na katere so razredi in posamezniki lahko povezani med seboj.
- Funkcijski pogoji (angl. Function terms): kompleksne strukture, ki nastanejo iz določenih razmerij, ki se lahko v določeni izjavi uporabljajo namesto posameznega izraza.
- Omejitve (angl. Restrictions): formalno navedeni opisi, kaj mora biti res, da bi lahko določena trditev bila sprejeta kot vhod.
- Pravila (angl. Rules): izjave v obliki »if-then« stavkov, ki opisujejo logične posledice, ki se lahko sklepajo iz trditve, če je le-ta v določeni obliki.
- Aksiomi (angl. Axioms): trditve (skupaj s pravili) v logični obliki, ki skupaj sestavljajo logično teorijo, ki jo antologija opisuje v svoji domeni. Ta definicija se razlikuje od definicije aksiomov v generativni slovnici (v ang. Generative grammar) in formalni logiki (angl. Formal logic). V teh disciplinah so aksiomi le izjave, uveljavljene kot znanje iz logike, brez dejstev, ki bi to logiko podpirali.
- Dogodki (angl. Events): spreminjanje atributov ali odnosov

V praksi razvoj ontologije zajema:

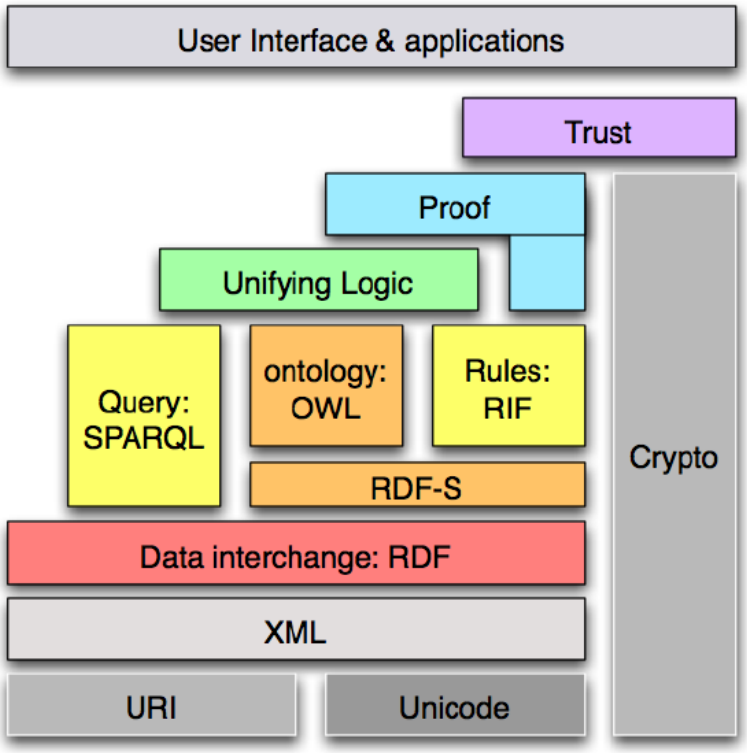
- Definicijo razredov v ontologiji
- Urejanje razredov v taksonomični (podrazred – nadrazred) hierarhiji
- Definicijo lastnosti in opis dovoljenih vrednosti
- Polnjenje vrednosti in ustvarjanje primerkov

Ko končamo z vsemi temi koraki, nam manjka še ustvarjanje posameznih primerkov razredov, da bi ustvarili bazo znanja neke domene.

### 2.1.2 Opis in razvoj tehnologij

#### 2.1.2.1 Spletni ontološki jezik

Spletni ontološki jezik (angl. Web ontology language, OWL)[2] je semantični spletni jezik, ki se uporablja pri predstavljanju bogatega in kompleksnega znanja o določenih stvareh, skupinah stvari in povezav med stvarmi. OWL je jezik, ki temelji na logiki, tako da se znanje, izraženo preko OWL, lahko obrazloži s pomočjo računalniških programov. Obrazložitev se dela, bodisi za preverjanje skladnosti tega znanja, bodisi da bi se implicitno znanje naredilo izrečno.




Slika 2.1: Struktura OWL jezikov

#### 2.1.2.2 Ogrodje za opis virov

Ogrodje za opis virov (angl. Resource description framework, RDF)[3] je standardni model za izmenjavo podatkov na spletu. RDF ima določene značilnosti, ki močno olajšajo združevanje podatkov, tudi v primerih, ko se osnovne sheme razlikujejo. RDF razširja povezovalno strukturo spleta prek uporabe enotnih označevalnikov virov (angl. Uniform resource identifier, URI) za označevanje, kakor povezave same, tako tudi oba konca te povezave. Ti trije deli tvorijo skupino, ki se običajno imenuje trojček (angl. triple). Uporabo tega preprostega modela omogoča izpostavljanje, mešanje in deljenje strukturiranih in napol strukturiranih podatkov med različnimi aplikacijami.

Takšna povezovalna struktura kot rezultat daje usmerjen in označen graf, kjer robovi predstavljajo poimenovano povezavo med dvema viroma, ki sta v bistvu vozlišči v grafu.

### 2.1.3 Spletna ontologija AJPES

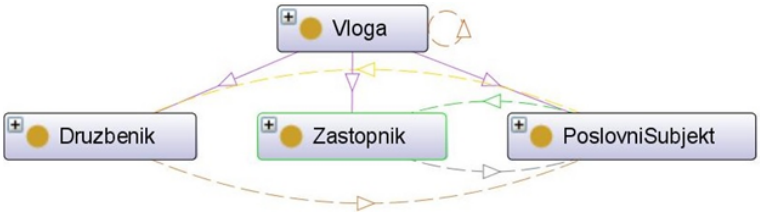
Kot smo že omenili, je bila za potrebe tega diplomskega dela ustvarjena ontologija, s pomočjo katere je bil izveden zajem podatkov iz AJPES-a. Ker je podatkovna množica AJPES-a zelo podrobna, je bilo treba napisati ontologijo v OWL, katerega primarna uporaba je definicija bolj zapletenih ontologij. Jezik OWL je sintaktično integriran v RDF in temelji na opisni logiki. Za serializacijo RDF-ja je bila uporabljena sintaksa RDF/XML.

Kot je razvidno iz Slike 2.2, smo definirali tri glavne razrede, iz katerih bodo izhajali podrazredi. Razreda *Naslov* in *Vloga* sta med seboj povezana z dvema lastnostma, ki sta med seboj inverzni.


Slika 2.2: Prvi nivo ontologije

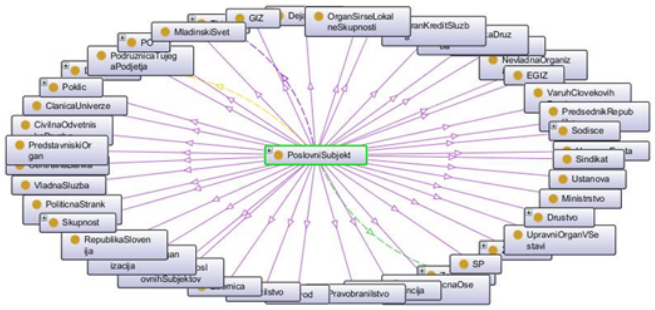
Razred *Vloga* je edini med tremi razredi na najvišjem nivoju, ki ima še

podrazrede. Njegovi podrazredi so vidni na spodnji sliki. Razredi *Druzbenik*, *Zastopnik* in *PoslovniSubjekt* tvorijo drugi nivo grafa, ki ga vidimo na Sliki 2.3.




Slika 2.3: Drugi nivo ontologije

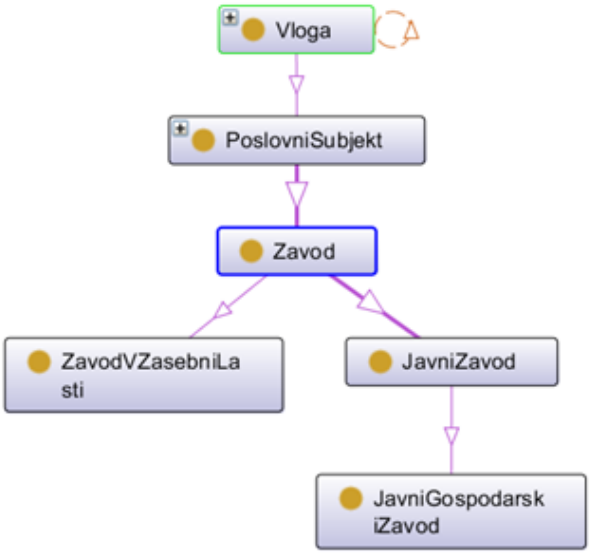
Od teh treh razredov je *PoslovniSubjekt* edini, ki ima od njega dedujoče razrede. Njegovi podrazredi so prikazani na Sliki 2.4 in vsi skupaj tvorijo tretji nivo grafa.




Slika 2.4: Tretji nivo ontologije

Graf ima še četrti in peti nivo. Razred *Zavod*, ki je del tretjega nivoja in je podrazred razreda *PoslovniSubjekt*, ima podrazred, ki se imenuje *JavniZavod*.

Peti nivo predstavlja podrazred razreda *JavniZavod*, ki se imenuje *JavniGospodarskiZavod*. Slednje je razvidno iz Slike 2.5.




Slika 2.5: Četrti in peti nivo ontologije

## 2.2 Grafi in omrežja

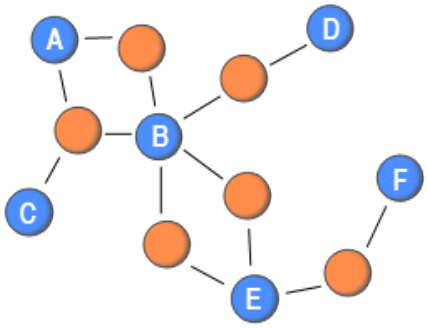
### 2.2.1 Osnovni pojmi

V teoriji grafov obstaja nekoliko ključnih pojmov. Eden izmed njih je *vozlišče* (angl. node). Vozlišče je osnovna enota, iz katere so sestavljeni grafi. Iz prejšnjega stavka lahko sklepamo, da množica vozlišč tvori *graf*. Vozlišča v enem grafu so med seboj povezana s *povezavami* (angl. edge), ki so lahko usmerjene ali neusmerjene. Pri usmerjenih povezavah sta natančno določeni vozlišči, ki sta vir in ponor. Pri neusmerjenih grafih sta lahko obe vozlišči, ki si delita povezavo, vir in ponor.

### 2.2.2 Eno in dvovrstna omrežja

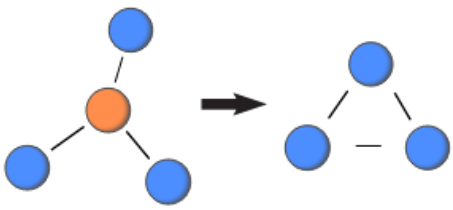
Omrežja so predstavitve sistemov, kjer so vozlišča med seboj povezana. Večina omrežij je definiranih kot enovrstna omrežja, kjer so vozlišča enega tipa oz. so si podobna med seboj, vendar pa obstajajo tudi dvovrstna omrežja[4], ki so sestavljena iz vozlišč dveh tipov. Takšna omrežja so nekaj posebnega, ker

so v njih povezave možne le med vozlišči različnih tipov. V takšnih omrežjih se obe množici vozlišč po navadi ločita glede na vplivnost določenega tipa, v procesu ustvarjanja povezav med vozlišči.




Slika 2.6: Primer dvovrstnega omrežja

Analize dvovrstnih omrežij so zelo redke. Po navadi se najprej transformirajo in se šele potem analizirajo. Transformiranje dvovrstnih omrežij v enovrstna se naredi s pomočjo metode, znane kot *projekcija*[4]. Projekcija dela tako, da izlušči vsa vozlišča enega tipa in jih med seboj poveže, če so v dvovrstnem omrežju imela vsaj eno skupno vozlišče drugega tipa. Z uporabo projekcije se potem jasno vidi, kateri tip vozlišč v omrežju je bolj vpliven, s tem pa tudi bolj primeren za analizo.

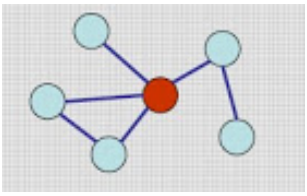


Slika 2.7: Projekcija dvovrstnega v enovrstno omrežje

### 2.2.3 Središčnost vozlišč omrežja

#### 2.2.3.1 Mere središčnosti glede na stopnjo

Morda je najbolj enostavna mera v nekem omrežju *stopnja vozlišča*[5]. Stopnja vozlišča predstavlja število neposrednih povezav, ki si jih to vozlišče deli z drugimi vozlišči. Da bi izračunali stopnjo določenega vozlišča, je treba samo preiskati njegovo najbližjo okolico. Čeprav je stopnja precej enostavna metrika, je lahko vir veliko informacij. V družabnem omrežju, kot je AJPES, lahko sklepamo, da večja stopnja določenega vozlišča pomeni povečan vpliv tega vozlišča v omrežju, olajšan dostop do informacije, ki jih imajo druga vozlišča, itn.



Slika 2.8: Primer vozlišča z visoko stopnjo

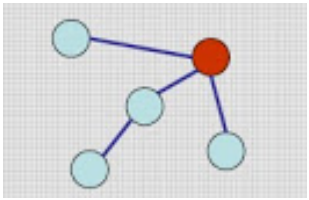
Stopnja se lahko izmeri za usmerjene, kakor tudi za neusmerjene grafe. Pri usmerjenih grafih iz splošne stopnje izhajata še *vhodna stopnja* (angl. in-degree) in *izhodna stopnja* (angl. out-degree), kot dodatni meritvi.

*Središčnost lastnega vektorja* (angl. Eigenvector centrality)[5] meri pomembnost določenega vozlišča v omrežju. Nanjo lahko gledamo kot naravni podaljšek stopnje vozlišča. K pomembnosti določenega vozlišča v omrežju prispeva dejstvo, da ima vozlišče neposredne povezave z drugimi vozlišči, ki so prav tako pomembna in vplivna v omrežju. V realnem svetu je po navadi neka oseba, ki pozna nekaj vplivnih oseb, bolj pomembna od tiste osebe, ki sicer pozna veliko ljudi, vendar ti nimajo nobenega vpliva v omrežju.

#### 2.2.3.2 Mere središčnosti glede na dostopnosti

*Bližinska središčnost* (angl. Closeness centrality)[5] se dokaj razlikuje od prej omenjenih metrikah. Bližinska središčnost meri povprečno razdaljo določenega

vozlišča, do vseh drugih v omrežju. Posplošeno, bližinska središčnost nam pove, koliko korakov v povprečju moramo narediti, da bi iz določenega vozlišča prišli do kakšnega drugega vozlišča v omrežju oz. grafu. Čeprav ima bližinska središčnost določene težave, katere zdaj ne bomo omenjali, se kot mera zelo pogosto uporablja pri analizi družabnih omrežij.



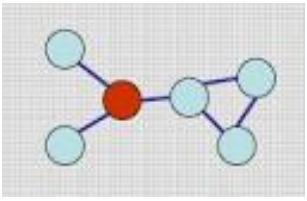
Slika 2.9: Primer vozlišča z visoko bližinsko središčnostjo

*Ekscentrična središčnost* (angl. Eccentricity, Eccentricity centrality)[5] je najdaljša izmed vseh najkrajših razdalj, ki jih lahko izračunamo za določeno vozlišče v omrežju. Ekscentrična središčnost je bolj uporabna, če je visoka. To je dejstvo, ki seveda ne pomeni, da ta metrika ni uporabna, če ima nižjo vrednost. Če ima določeno vozlišče visoko vrednost za ekscentrično središčnostjo to pomeni, da so vsa ostala vozlišča v njegovi bližini. Veljavna je tudi obratna trditev, da nižja vrednost narekuje obstoj vsaj enega, vendar lahko tudi več vozlišč in vse njihove neposredne sosede, ki niso v bližini tega vozlišča.

#### 2.2.3.3 Mere središčnosti glede na vmesnosti

*Središčnost vmesnosti* (angl. Betweenness centrality)[5] je mera, ki nam pove, koliko pogosto je določeno vozlišče na poti med drugimi vozlišči. Ideja središčnosti vmesnosti je, da imajo tista vozlišča, ki so neke vrste most med katerimkoli parom drugih vozlišč v omrežju, več informacij in so tako bolj vplivna v celotnem omrežju. Če bi se vozlišče, ki ima visoko središčnost vmesnosti odstranilo iz omrežja, bi se pretok informacij v omrežju v veliki meri spremenil.

Vmesnost je metrika, ki ni odvisna od usmerjenosti grafa. Res je, da je bil



Slika 2.10: Primer vozlišča z visoko središčnostjo vmesnosti

ob njeni definiciji uporabljen usmerjen graf, vendar pozneje se je dokazalo, da je uporabna prav tako pri neusmerjenih grafih.

### 2.2.4 Metode analize povezav

*PageRank*[5] je algoritem, ki je vsem računalničarjem zelo dobro znan. To je algoritem, ki ga je razvil eden od ustanoviteljev Google-a, Larry Page in se uporablja za razvrščanje spletnih strani, ki so del nabora rezultatov pri določenem spletnem iskanju. Pri tem se vsebina spletne strani sploh ne upošteva, upoštevajo se le povezave, ki jih imajo spletne strani z drugimi spletnimi stranmi. Pri omrežjih, PageRank razvršča vozlišča glede na njihovo pomembnost v omrežju, pri čemer se upoštevajo povezave vozlišč.

*Kazalo* (angl. Hub) in *vir* (angl. Authority)[5] sta dve metriki, ki se izračunata s pomočjo HITS algoritma. V teoriji je HITS (angl. Hyperlink-induced topic search) elegantna rešitev, ki pove veliko o središčnosti določenega vozlišča, čeprav se v praksi ne uporablja veliko.

Vozlišče je vir z visoko vrednostjo, če na njem kaže veliko vozlišč, ki so kazala. Kar se pa tiče vrednosti za kazala, je ta za določeno vozlišče velika, če je vozlišče povezano z veliko vozlišči, ki so vir.

## 2.3 Podatkovno rudarjenje

*Podatkovno rudarjenje*[6] je veja v računalništvu, kjer se kot rezultat pojavljajo vzorci[1]. Podatkovne množice se analizirajo z različnimi metodami, ki imajo značilnosti strojnega učenja, umetne inteligence in statistike.

### 2.3.1 Osnovni pojmi

Pri podatkovnem rudarjenju je navada, da se podatkovna množica razdeli na *učno* in *testno*. Učna množica je tista, na kateri napovednega modela učimo, kako naj se obnaša. Testna množica je pa tista, na kateri preizkušamo ali se model resnično obnaša tako, kot smo ga naučili.

### 2.3.2 Klasifikacija in regresija

V statistiki se zelo pogosto uporablja *klasifikacija*[7]. Klasifikacijo lahko označimo kot zmanjševalno tehniko, ker je njena največja značilnost razdeljevanje podatkov iz določene podatkovne množice v več razredov. Pomembno je omeniti, da je razred v bistvu diskretna spremenljivka.

*Regresija*[8] je statistični proces, ki se uporablja pri ocenjevanju razmerja med spremenljivkami. Velja poudariti, da se regresija zelo pogosto uporablja pri napovedovanju podatkov. Za razliko od klasifikacijskega pristopa morajo biti spremenljivke, ki se uporabljajo v procesu regresije zvezne.

V diplomski nalogi smo uporabili klasifikacijo in regresijo. Konkretno v našem primeru, klasifikacija napoveduje ali sploh pride do spremembe, regresija pa napoveduje, kako velika je bila sprememba. Čeprav sta oba pristopa različna, je bilo smiselno uporabiti oba in ugotoviti, kako se bosta obnašala pri naših podatkih.

[1]Ponavljajoči se niz podatkovnih elementov ali druga vrsta zakonitosti v podatkih, ki navadno omogoča napovedovanje vrednosti nekaterih podatkovnih elementov v velikih podatkovnih množicah

### 2.3.3 Metode podatkovnega rudarjenja

#### 2.3.3.1 Klasifikacijski algoritmi

Pri testiranju klasifikacije smo uporabili štiri algoritme, ki so na voljo za uporabo v programskem paketu Orange. To so *Naivni Bayesov* (angl. Naive Bayes, NB)[9], *K-najbližje sosede* (angl. k nearest neighbours, kNN)[10], *Klasifikacijsko drevo* (angl. Classification tree, CT)[11] in *Naključni gozd* (angl. Random Forest, RF)[12][13].

Naivni Bayesov algoritem na osnovi podane razredne spremenljivke predpostavlja, da je odsotnost ali prisotnost določene značilnosti nepovezana z odsotnostjo ali prisotnostjo kakršne koli druge značilnosti.

Algoritem kNN napoveduje vrednost (v našem primeru) vozlišča s tem, da upošteva le najbližje testne primere. Je najbolj enostaven od vseh algoritmov, ki se uporabljajo pri strojnem učenju. Obstaja različica kNN algoritma, ki je prilagojena za uporabo regresije.

Klasifikacijska drevesa se uporabljajo takrat, kadar so nam razredi učne množice znani vnaprej. Obstaja pa več načinov za pridobivanje razredov v učnih podatkovnih množicah. Naš način bomo pojasnili v nadaljevanju.

Random Forest je algoritem, kjer se hkrati uporablja večje število klasifikacijskih dreves. Vsako drevo je odvisno od vrednosti naključno vzorčenega vektorja. V različnih drevesih se uporabljajo različni vektorji, ki pa morajo biti enako porazdeljeni.

#### 2.3.3.2 Regresijski algoritmi

Pri testiranju regresije smo prav tako uporabili 4 algoritme, in sicer *kNN*[14], *regresijsko drevo* (angl. Regression tree)[11], *Random Forest regresijskega algoritma*[12][13] in *SVM* (angl. Support vector machine) regresijskega algoritma[15].

kNN algoritem je v osnovi enak kot tisti, ki smo ga opisali v Poglavju 2.3.3.1, le da je prilagojen za napovedovanje zveznih spremenljivk.

Regresijska drevesa so podobna klasifikacijskim drevesom s tem, da se pri njihovi uporabi ne uporabljajo razredi. Razdelitev se dela v skladu z *Metodo*

*najmanjših kvadratov.*

Pri regresijskem napovedovanju, algoritem Random Forest uporablja regresijska drevesa, namesto klasifikacijska. Sicer pa je princip algoritma pri klasifikacijskem in pri regresijskem napovedovanju enak.

SVM je algoritem, ki se uporablja na večih področjih. Bistvo različice, ki se uporablja pri regresijskem napovedovanju je, da model, ki se dobi kot rezultat, ni zgrajen na celotno učno množico, ampak za majhen del le-te. Razlog za to je, da se ignorirajo tisti učni podatki, ki so po vrednosti približno enaki kot tisti, ki jih je model napovedal.

#### 2.3.3.3 Algoritmi za razvrščanje atributov

Pravilna izbira atributov, ki najbolj vplivajo na končni rezultat je zelo pomembna pri strojnem učenju. Uporaba atributov, ki na proces učenja ne vplivajo veliko, bo rezultirala z modelom, ki dela slabo. V Orange-u je implementiranih veliko algoritmov, ki razvrščajo algoritme po njihovi pomembnosti.

Najbolj pogosto uporabljeni algoritmi za razvrščanje atributov pri klasifikacijskih problemih so *Information gain*[16], *Gini index*[17], *MSE* (angl. Minimum squared error)[18], *ReliefF*[19], itd.

Pri regresijskih problemih se največkrat uporabljajo algoritmi *MAE* (angl. Mean absolute error)[20], *MSE* (angl. Mean squared error)[18], *RReliefF*[19], itd.

#### 2.3.3.4 Metrike pri regresijskem napovedovanju

Obstaja veliko metrik, na osnovi katerih lahko ocenimo natančnost delovanja regresijskih algoritmov, ki smo jih omenili v Poglavju 2.3.3.2.

Med najbolj pogosto uporabljenimi v takšnih primerih sta *Povprečna absolutna napaka* (angl. Mean absolute error, MAE)[20] in *Relativna absolutna napaka* (angl. Relative absolute error, RAE)[20].

*MAE* meri koliko blizu je napovedana do realne vrednosti. Podana je s formulo:

$$MAE = \frac{1}{n} \times \sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n} \times \sum_{i=1}^{n} |e_i|$$

kjer je $f_i$ napovedana vrednost, $y_i$ je dejanska vrednost, $n$ pa število primerkov. Posplošeno, MAE je povprečje razlik med napovedano in dejansko vrednostjo.

*RAE* je enaka metrika kot zgoraj omenjene MAE, le, da se izraža v odstotkih.

#### 2.3.3.5 Metrike pri klasifikacijskem napovedovanju

Metrike za ocenjevanje klasifikacijskih algoritmov, opisanih v Poglavju 2.3.3.1 se dokaj razlikujejo z merami za ocenjevanje regresijskih algoritmov. V nadaljevanju bomo podali kratko razlago metrik, ki so bili uporabljeni pri interpretaciji klasifikacijskega napovedovanja.

*Klasifikacijska točnost* (angl. Classification accuracy, CA)[21] je mera, ki odraža razmerje med pravilno klasificiranimi in vsemi primeri.

*Občutljivost* (angl. Sensitivity)[22] in *specifičnost* (angl. specificity)[22] sta meritvi, ki se skorajda vedno uporabljata skupaj. Občutljivost meri verjetnost t.i. *resnično pozitivnih* (angl. true positives). Posplošeno, podaja odstotek tistih, katerim je bila napovedana sprememba in se jim je sprememba zares zgodila. Specifičnost pa meri odstotek t.i. *resnično negativnih* (angl. true negatives), oz. odstotek tistih, katerim ni bila napovedana sprememba in se jim sprememba ni zgodila.

Občutljivost je podana s formulo:

$$sensitivity = \frac{TP}{TP + FN}$$

kjer je $TP$ število tistih, katerim je bila napovedana sprememba in se jim zgodila, $FN$ pa predstavlja število tistih, katerim je bila napovedana sprememba, vendar se jim ni zgodila.

Specifičnost pa meri ravno obratno. Podana je s formulo:

$$specificity = \frac{TN}{TN + FP}$$

kjer je $TN$ število tistih, katerim ni bila napovedana sprememba in sprememba se jim ni zgodila, $FP$ pa predstavlja število tistih, katerim ni bila napovedana sprememba, vendar se sprememba zgodila.

*Brier ocena* (angl. Brier score) meri natančnost verjetnostnih napovedi. Če smo malo bolj podrobni, Brier ocenjuje povprečno kvadrirano napako verjetnostjo dogodka in resničen rezultat napovedovanja o omenjenem dogodku. Podana je s formulo:

$$BS = \frac{1}{N} \sum_{t=1}^{N} (f_t - o_t)^2$$

kjer je $f_t$ napovedana verjetnost, da se bo napovedan dogodek res zgodil, $o_t$ je pa rezultat napovedovanja, ki je lahko 0 (če se ne dogodek ne zgodi) in 1 (če se dogodek zgodi), $N$ pa je število vseh primerkov.

*Površina pod ROC krivuljo* (angl. Area under the ROC curve, AUC)[23][24] je mera, ki je sposobna nekoliko boljše kot ostalih meritvah ravnati in oceniti klasificiranje, kjer podatki niso enako razdeljeni med razredov.

## 2.4 Uporabljene knjižnice in orodja

Pri izdelavi diplomskega dela smo uporabili dve zunanji knjižnici, kateri je bilo treba vključiti v uporabljenem razvojnem okolju. Poleg tega smo uporabili še programski paket Orange, ki se uporablja pri strojnem učenju in rudarjenju podatkov.

### 2.4.1 OWL API

Naša ontologija je bila napisana v *OWL*. OWL je družina ontoloških jezikov oz. družina jezikov za predstavljanje znanja, pridobljenega iz neke domene. Kaj pa zajema baza znanja, smo povedali v Poglavju 2.1.1. Vsi jeziki iz OWL družine temeljijo na formalni semantiki in na serializaciji o semantičnem spletu, serializacije pa običajno temeljijo na RDF, XML ali na obeh.

OWL programski vmesnik[26] smo načeloma uporabljali za ravnanje z ontologijami in s posamezniki. V OWL programskem vmesniku je ontologija *(OWLOntology)* definirana kot vmesnik, ki modelira množico logičnih in nelogičnih povezav med razredi in posamezniki *(OWLAxioms)* z imenom *(angl. Internationalzed resource identifiers, IRI)*, (po izbiri) fizično lokacijo in priročnimi metodami za pridobivanje takšnih aksiomov.

Del, ki je povezan z uporabo OWL programskega vmesnika, je šel brez večjih težav. Manjše težave so bile le na začetku, dokler se nisem dovolj dobro seznanil s tematiko in s tem, kako in koliko dobro deluje programski vmesnik. Edina pomanjkljivost, ki jo lahko izpostavim, je nezmožnost uporabe dveh ontologij, ki imata isto ime. Omenjena pomanjkljivost sicer ni veliko vplivala na čas izvajanja, vendar je smiselno, da se je omeni.

### 2.4.2 Gephi Toolkit

Podatki, ki so bili obdelani s pomočjo OWL programskega vmesnika, so bili potem organizirani v grafih. Za ta namen smo uporabljali orodjarno *Gephi*[27]. Zgrajen je po zgledu namiznega programa Gephi, ki se uporablja za interaktivno prikazovanje in raziskovanje vseh vrst omrežij in kompleksnih dinamičnih in hierarhičnih grafov. Gephi programski vmesnik je zmogljiv in z njim se lahko dela prav vse, kar lahko počnemo z namiznim programom Gephi. Za potrebe tega diplomskega dela smo ga uporabljali za ustvarjanje, filtriranje, uvažanje in izvažanje grafov kakor tudi računanje metrik, ki so značilne za vozlišča in povezave v grafih.

Edina večja težava, ki se je pojavljala skozi celoten proces pisanja te diplomske naloge, je bila pomanjkljivost Gephi programskega vmesnika, ki je zatajil pri izvažanju podatkov o izračunanih metrikah. Pomanjkljivost je prišla do izraza, ko smo računali metrike za grafe, ki so imeli vsaj 5.000 vozlišč in 6.000 povezav. Tako je Gephi programski vmesnik pri določenih metrikah dejansko vrednost povečal za vsaj $10^{12}$, kar pa je seveda narobe. Treba je bilo izvožen graf odpreti z namizno različico Gephi-ja in ročno popraviti vse podatke. Takšno delo je zahtevalo veliko časa in veliko potrpežljivosti, pa

tudi pozornost pri popravljanju podatkov. Napačni podatki bi zagotovo bili razlog za morebitne napake v analizi.

### 2.4.3 Orange

*Orange*[28] je programski paket, ki se uporablja za rudarjenje in analizo podatkov, kakor tudi pri strojnem učenju. V tej diplomski nalogi smo ga uporabljali za testiranje natančnosti vsake variante. Uporabljali smo več algoritmov, da bi dejansko ugotovili, kateri izmed njih dela najboljše oz. daje najboljše rezultate. Na podlagi dobljenih rezultatov smo se odločili, na katero izmed vseh testiranih variant se bomo osredotočili.

Pri uporabljanju programskega paketa Orange se je večkrat zgodilo, da je le-ta naenkrat prenehal z delovanjem. Nekajkrat je nepričakovano »zmrzovanje« programa povzročilo izgubo neshranjenega dela.

# Poglavje 3

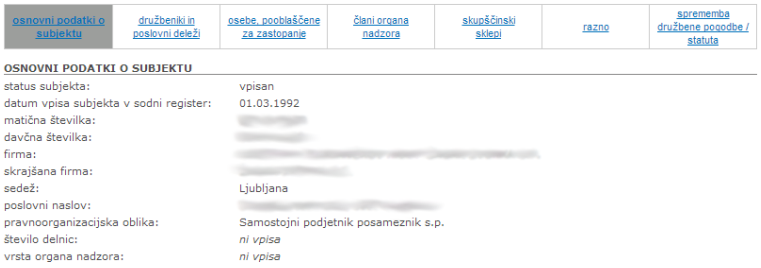
# Napovedovanje lastništva iz omrežij družbenikov

## 3.1 Podatkovna zbirka AJPES

Kot vir podatkov v tej diplomski nalogi smo uporabili *Agencijo Republike Slovenije o javnopravnih evidencah in storitev (AJPES)*. AJPES je ustanova, ki je nastala v letu 2002, po razgraditvi Službe družbenega knjigovodstva Republike Slovenije. Glavna naloga AJPES-a je zbiranje, posredovanje in obdelava finančnih poročil poslovnih osebkov v Republiki Sloveniji. Poleg finančnih podatkov, hranijo se tudi ustanovitveni podatki posameznih podjetij, družbeniki v posameznih poslovnih podjetjih in njihov lastniški delež, pooblaščenih oseb, itn.

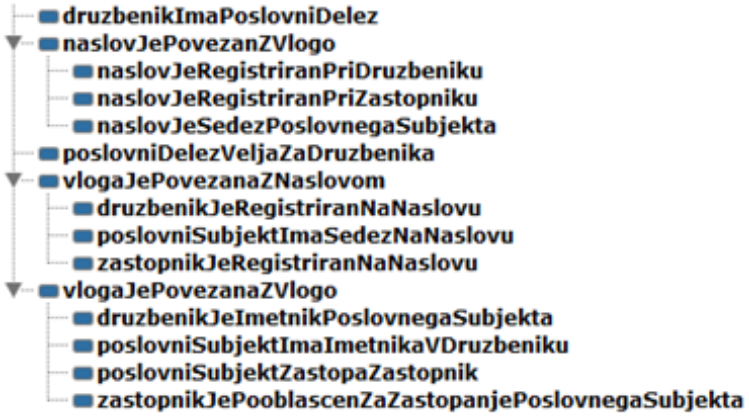
Podatki so bili zajeti s pomočjo ontologije, ki smo jo začeli opisovati v Poglavju 2.1.3. Velja poudariti, da zajeti podatki predstavljajo le neznaten del AJPES-ove podatkovne baze[1]. Zdaj, ko imamo vsaj vizualno podobo (glej Sliko 3.1), katere podatki se hranijo na AJPES-u, lahko podrobneje opišemo ontologijo.

[1]Uporabnik lahko naenkrat uporablja le neznatne dele podatkovnih baz (kakovostno in/ali količinsko) oziroma v obsegu, ki je predviden v okviru razpoložljivih spletnih aplikacij. Celotni pogoji uporabe so dostopni na `http://www.ajpes.si/Pogoji_uporabe`




Slika 3.1: Organizacija zapisa na AJPES-ovi spletni strani. Podatki so anonimizirani, zaradi varovanja zasebnosti

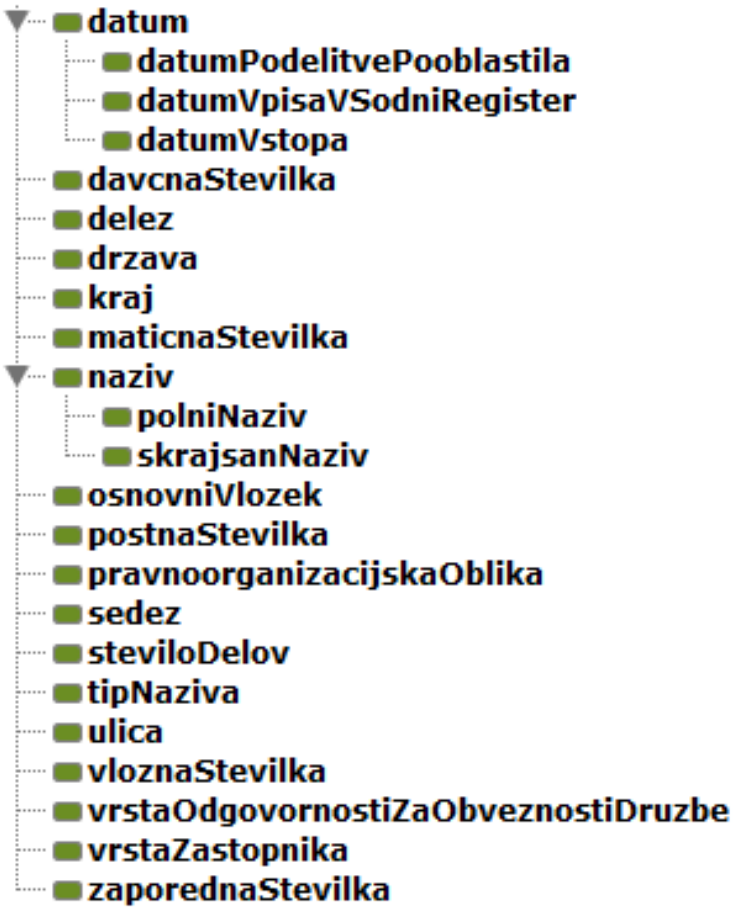
Kar se tiče atributov v ontologiji, sta bili definirani dve vrsti. To so *razredni* in *podatkovni atributi*. Kot sami imeni kažeta, se nanašata zaporedoma na razrede in na podatke. Obstaja 5 glavnih razrednih atributov, iz katerih izhajajo drugi pod-atributi. Na Sliki 3.2 so vidne glavne razredne lastnosti in iz njih izhajajoči pod-atributi.




Slika 3.2: Razredne lastnosti definirane ontologije

Razredne atribute opisujemo z domeno in obsegom. *Domena* je razred, iz katerega ta atribut izhaja, *obseg* pa je razred, s katerim domena tvori presek, da bi povezala dva razreda. Posplošeno, domena je vir, obseg je pa ponor razrednega atributa. Pri predstavljanju prvega nivoja (glej Sliko 2.2) ontologije smo omenili, da sta dva razreda med seboj povezana z inverznima atributoma. To je značilnost razrednih atributov, da lahko obstajata atributa, ki

sta med seboj nasprotujoča. Takšna sta atributa *druzbenikJeRegistriranNaNaslovu* in *naslovJeRegistriranPriDruzbeniku*, ki povezujeta razreda *Druzbenik* in *Naslov*.




Slika 3.3: Podatkovne lastnosti definirane ontologije

Na Sliki 3.3 je prikaz vseh podatkovnih atributov. Lahko opazimo, da so tudi pri podatkovnih atributih dedujoči atributi. Takšna je, na primer, skupina atributov datum. To skupino sestavljajo atributi *datumPodelitvePooblastila, datumVpisaVSodniRegister* in *datumVstopa*. V takih primerih si skupina pod-atributov deli obseg (tip atributa), različna pa je domena. Tako ima atribut *datumPodelitvePooblastila* svojo domeno v razredu *Zastopnik*, atribut *datumVpisaVSodniRegister* v razredu *PoslovniSubjekt* in posledično

v vseh njegovih podrazredih, atribut *datumVstopa* pa ima domeno v razredu *Druzbenik*.

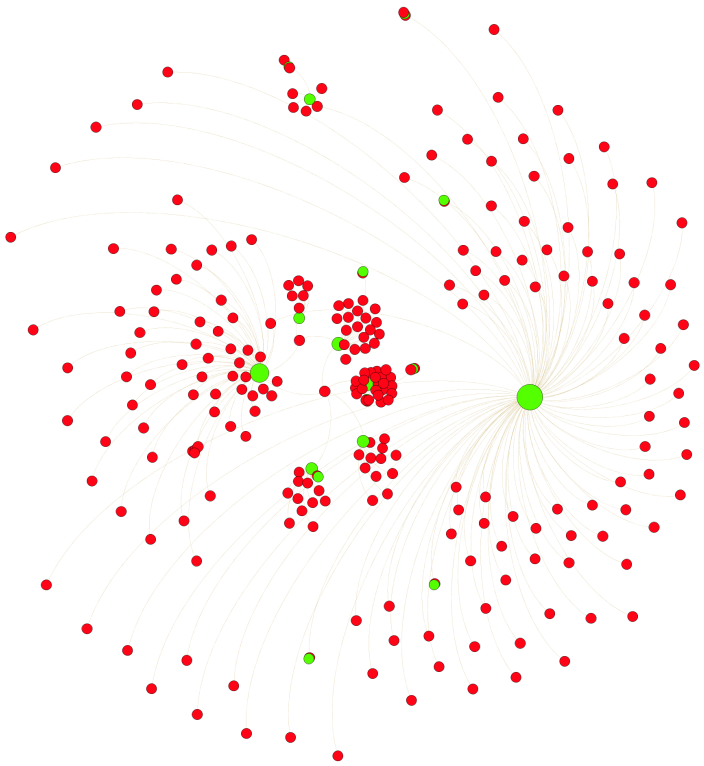
V Poglavju 2.1.3 smo se seznanili z razredi, ki so definirani v ontologiji. Kot se lahko vidi iz števila zavihkov o enem zapisu (glej Sliko 3.1), je podatkov veliko in jih v veliki meri ni bilo mogoče smiselno povezati med seboj iz preprostega razloga, ker so to zgolj informativni podatki o poslovnih osebkih. Podatki o družbenikih, poslovnih deležih in poslovnih osebkih so bili edini podatki, med katerimi bi lahko obstajala kakšna povezava, iz katere bi bilo mogoče potegniti kakšno ugotovitev, ki bi dejansko bila uporabna še v realnem svetu. Zaradi tega analiza v tej diplomski nalogi temelji prav na teh podatkih.

## 3.2 Omrežja družbenikov in podjetij

OWL programskega vmesnika smo uporabljali za obdelavo pridobljenih podatkov. Najprej smo potrebovali vse primerke razredov *PoslovniDelez, Druzbenik* in *PoslovniSubjekt*. Z nadaljnjo obdelavo vseh pridobljenih primerkov, smo pridobili družbenike in njim pripadajoča podjetja. Družbeniki in podjetja predstavljajo vozlišča, ki jih je treba med seboj povezati, da bi dobili celotno omrežje. Velja poudariti, da ima vozlišče tipa družbenik lahko povezavo le z vozliščem različnega tipa, in obratno. Vozlišča smo med seboj povezali na podlagi dejstva, da ima družbenik delnice v določenem podjetju. Tako smo dobili omrežja, kjer so vozlišča različnega tipa.

Značilnost tako dobljenih omrežij je bila zelo šibka povezanost, saj je večina vozlišč imela stopnjo 1. Analiza takšnih omrežij ne bi bila smiselna, ker bi bili dobljeni rezultati porazni. Vsem dobljenim omrežij smo s pomočjo Gephi programskega vmesnika dodali filter, da bi iz njih dobili največjo povezano komponento.

Največja povezana komponenta določenega grafa je v bistvu podmnožica celotnega grafa, kjer so vsa vozlišča neposredno ali posredno povezana med seboj. Skratka, iz vseh ostalih vozlišč je možno priti do vsakega drugega



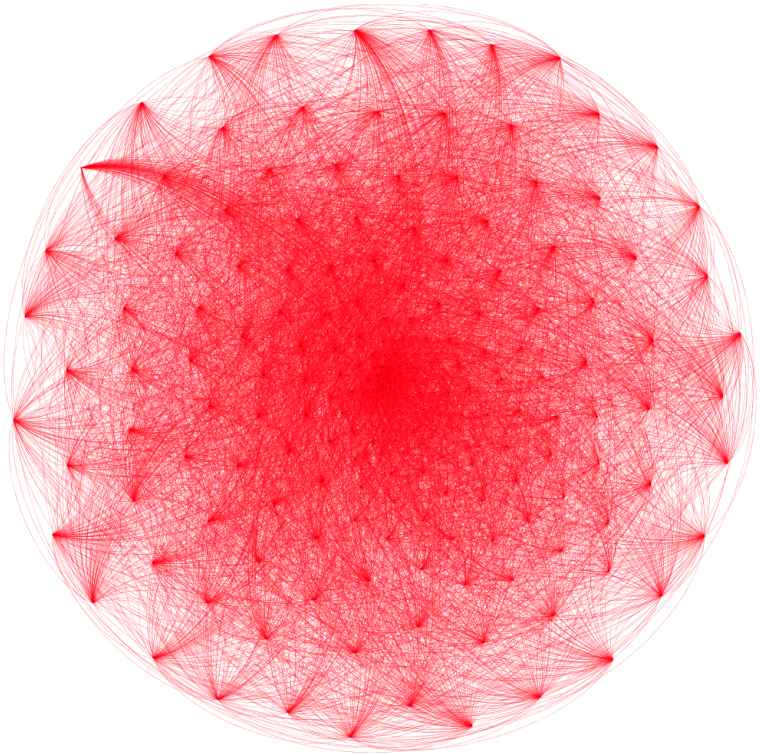
Slika 3.4: Največja povezana komponenta, izluščena iz dvovrstnega omrežja. Z rdečo barvo so označeni družbeniki, z zeleno pa podjetja

vozlišča v omrežju. Neusmerjenost grafa nam je dodatno olajšala stvar.

V Poglavju 2.2.2 smo omenili, da je analiza dvovrstnih omrežij zelo redka praksa. Razlog za to je, da je večina metrik za analizo omrežij definiranih za enovrstna omrežja. Le nekateri so bili na novo definirani in uporabljeni pri analiziranju dvovrstnih omrežij. Mi smo se tega držali in analize dvovrstnih omrežij nismo izvedli.

Analizo smo osredotočili na enovrstna omrežja. Enovrstna omrežja smo pridobili s projekcijo dvovrstnih omrežij. Tako smo iz enega dvovrstnega omrežja dobili dve enovrstni. Prvo omrežje je bilo sestavljeno iz vozlišč tipa družbenik, drugo pa iz vozlišč tipa podjetje. V omrežju družbenikov sta bili posamezni vozlišči med seboj povezani, če sta si v dvovrstnem omrežju delila vozlišče tipa podjetje. Število vozlišč v omrežju na Sliki 3.5 je 244, med

vozliščih pa obstaja 7.760 povezav.

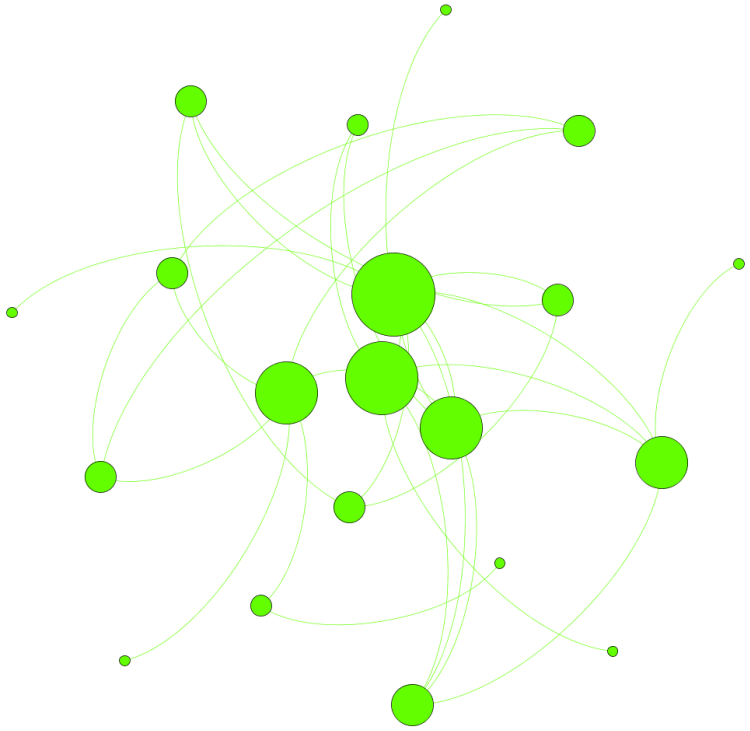


Slika 3.5: Enovrsten graf sestavljen iz družbenikov

Za omrežja podjetij pa velja obratna trditev: dve vozlišči tipa podjetje sta bili med seboj povezani, če sta si v dvovrstnem omrežju delila vozlišče tipa družbenik. Število vozlišč za enovrstno omrežje podjetij na Sliki 3.6 je 20, med temi 20-imi vozlišči pa obstaja 32 povezav.

Iz Slik 3.5 in 3.6 lahko sklepamo, da je enovrstno omrežje družbenikov večje in bolj povezano v primerjavi z omrežjem podjetij. Takšen izhod je bil tudi pričakovan, saj so družbeniki številčnejši kot podjetja.

Sliki nas pripeljeta do še enega sklepa. V Poglavju 2.2.2 smo omenili, da se s projeciranjem ugotovi, kateri tip vozlišč v dvovrstnem omrežju je vplivnejši. V našem primeru, to je očitno enovrstno omrežje družbenikov. Analiza enovrstnega omrežja, ki je sestavljeno iz vplivnejšega tipa vozlišč, je primernejša za analizo v primerjavi z analizo omrežja, ki je sestavljeno iz



Slika 3.6: Enovrsten graf, ki vsebuje le podjetja

manj vplivnih vozlišč. Analizo smo vseeno izvedli na obeh tipih omrežij, da bi dokazali prej omenjeno trditev.

## 3.3 Napovedovanje sprememb lastništva

Zadnje čase je analiza družbenih omrežij postala ključna tehnika in se uporablja na raznovrstnih področjih. Poleg povečane uporabe v akademskih krogih, se analiza družbenih omrežij veliko več kot prej uporablja za praktične namene, kot na primer za odkrivanje raznovrstnih goljufij, teroristične dejavnosti, itd. Pri tem se uporabljajo številne meritve, ki so značilne za omrežja.

Glavna stvar, na katero smo se osredotočili v tej diplomski nalogi je bila, kako in čim bolj učinkovito napovedati spremembo stopnje vozlišč v omrežju. Napovedujemo spremembo stopnje, ker je tako predstavljena sprememba lastništva v našem omrežju. Za dosego tega cilja smo uporabljali metrike

središčnosti, ki so opisane v Poglavju 2.2.3. Mere središčnosti se uporabljajo pri kvantificiranju pomembnosti in vplivnosti posameznih vozlišč ali skupine vozlišč v omrežju.

Pomembno je omeniti, da smo napovedovanje delali na več časovnih oknih. Časovno okno se nanaša na datum, ko je družbenik vstopil v podjetje. Za vpeljavo časovnega filtra, je bilo potrebno upoštevati podatkovni atribut *datumVstopa* (glej Sliko 3.3). Spremljali smo podatke od začetka leta 1992 do konca leta 2012. Začeli smo z 2-letnim časovnim oknom, končali smo pri 5-letnem. Morali smo biti pozorni na to, da je podatkovna množica dovolj velika in da je število sprememb o vsakem vozlišču dovolj veliko, da bi se lahko model dovolj dobro naučil. Nadaljnje povečevanje časovnega okna bi sicer poskrbelo za več testnih primerov, vendar nekateri od njih ne bi bili uporabni, zaradi premajhnega števila sprememb za vsako vozlišče.

V Poglavju 2.3.1 smo omenili, kako se pri strojnem učenju podatkovna množica običajno razdeli na učno in testno. Učno množico smo uporabili, da bi se naučili, kako se stopnja spreminja. Testna množica pa je bila uporabljena za to, da bi z uporabo različnih pristopov poskušali ugotoviti ali smo se pravilno naučili napovedovati spremembe. Zaradi časovnega okna so bili naši podatki razdeljeni. Zaradi tega je bilo potrebno vpeljati atribut *period*. Ta nam je pomagal pri razdeljevanju podatkovne množice na učne in testne. Na primer, pri povečevanju časovnega okna za 2 leti, časovno obdobje 1992-2012 smo razdelili na 10 delov, s čimer smo dobili 10 datotek. Vsako datoteko smo ustrezno označili z uporabo atributa *period*, ki v tem primeru ima vrednosti 0-9. Na koncu smo vse datoteke, ustvarjene na istem časovnem oknu združili v eno. Podatki, kjer je bil atribut *period* $<= 8$ smo premaknili v učno, vse ostale pa v testno množico.

Spremembo stopnje smo napovedovali na tri načine. Prvi način je bil klasifikacijsko napovedovanje oz. ali sploh pride do spremembe ali ne. Za izvedbo klasifikacijskega napovedovanja smo v podatkovni množici dodali atribut *change_ or_ no*, ki ima lahko dve vrednosti oz. razreda. Prvi razred, ki smo ga definirali, je bil 0. Vrednost 0 napoveduje, da spremembe stopnje

ne bo. Drugi razred je 1, ki napoveduje, da se bo v naslednjem obdobju zgodila sprememba stopnje. Uporabljeni so bili klasifikacijski algoritmi, ki smo jih opisali v Poglavju 2.3.3.1.

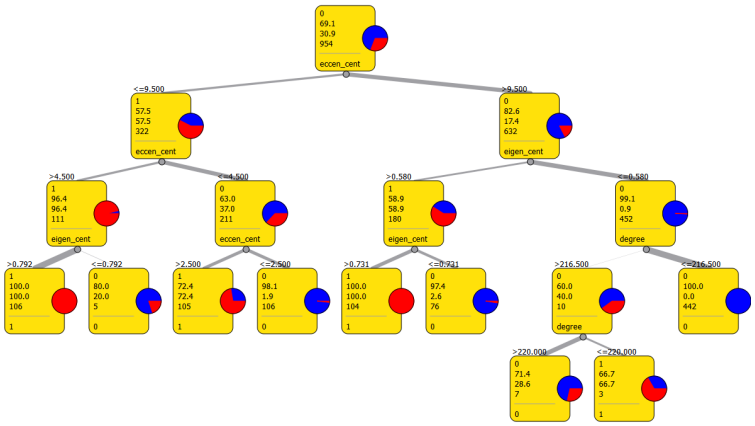
Pri drugem načinu smo napovedovali spremembo stopnje v odstotkih. Vpeljali smo atributa *percentage*, ki predstavlja spremembo stopnje vozlišča med naslednjim in trenutnim časovnim obdobjem, v odstotkih. Pri preizkušanju tega načina so bili uporabljeni regresijski algoritmi, ki smo jih že opisali v poglavju 2.3.3.2.

Tretji način je bil napovedovanje številčne spremembe stopnje. Za pomoč pri testiranju smo podatkovni množici dodali atribut *delta*, ki predstavlja velikost spremembe med stopnjama v naslednjem in trenutnem časovnem obdobju. Pri testiranju so bili uporabljeni isti algoritmi, kot pri napovedovanju spremembe stopnje v odstotkih.

## 3.4 Gradnja napovednega modela

Rezultat vseh analitskih pristopov, opisanih v Poglavju 3.3 je napovedni model, s katerim je mogoče napovedovati spremembo stopnje vozlišča.

Napovedni model je v bistvu klasifikacijsko drevo, ki na podlagi pravil v vozliščih, napoveduje ali bo prišlo do spremembe v stopnji vozlišča. Na Sliki 3.7 je prikazan enostavni napovedni model.




Slika 3.7: Enostavni napovedni model

# Poglavje 4

# Rezultati in diskusija

V Poglavju 4 bomo predstavili dejanske rezultate izvedene analize.

## 4.1 Izbira ustreznih metrik

Pri analiziranju omrežij je zelo pomembno dobro izbrati atribute, ki se bodo uporabili pri procesu napovedovanja. V poglavju 2.3.3.3 smo se seznanili z nekaterimi od najbolj pogosto uporabljenih algoritmov za ocenjevanje pomembnosti atributov.

Našo odločitev smo zasnovali na rezultate, dobljene z uporabo algoritma *ReliefF*, ki je bil razvit prav na *Fakulteti za računalništvo in informatiko*, s strani prof. dr. Igorja Kononenka in izr. prof. dr. Marka Robnika Šikonje.

Upoštevali smo rezultate 24. podatkovnih množic. V programskem paketu Orange obstaja gradnik po imenu *Rank*, ki atribute razvršča po svoji uporabnosti za naloženo podatkovno množico. Naša odločitev je bila zasnovana na rezultate prej omenjenega gradnika. Pogledali smo, kako so bili atributi razvrščeni po svoji uporabnosti. V večini primerov je bila središčnost lastnega vektorja za podatkovno množico najbolj uporaben atribut. Zaradi tega smo središčnost lastnega vektorja izbrali kot enega od atributov, na katerega se bomo zanašali pri preizkušanju različnih variant.

Drugi atribut, ki smo ga izbrali, je bila stopnja vozlišča. V 90% vseh

primerov je stopnja vozlišča bila po uporabnosti, uvrščena na drugo mesto. Tudi če ne bi bila, bi bilo smiselno uporabiti stopnjo vozlišča, saj je to atribut, ki ga v bistvu napovedujemo.

Tretji izbrani atribut je bil ekscentrična središčnost. V povprečju je bil ekscentrična središčnost po uporabnosti tretje uvrščen atribut. Tako smo, namerno ali nenamerno izbrali atribute, ki glede na to kaj merijo, pripadajo dvema različnima skupinama (glej Poglavje 2.2.3.1).

## 4.2 Napovedovanje lastništva družbenikov in podjetij

V Poglavju 3.2 smo omenili, da se pri projekciji dvovrstnih omrežij na enovrstne, analiza običajno izvaja za tisto enovrstno omrežje, ki je sestavljeno iz bolj vplivnih vozlišč. Videli smo, da je to v našem primeru enovrstno omrežje, ki je sestavljeno iz družbenikov. V tem poglavju bomo dokazali, da ni smiselno analizirati enovrstnih omrežij, ki so sestavljena iz manj vplivnih vozlišč.

Številke, ki bodo prikazane v Tabeli 4.1 so rezultati analize, ki je bila izvedena na 5-letnem časovnem oknu. Nismo izbrali manjšega časovnega okna, ker v podatkovnih množicah podjetij, zajetih na manjšem časovnem oknu ni bilo dovolj podatkov, da bi ustvarjen napovedni model bil dovolj dober. Tudi na 5-letnem časovnem oknu podatkovna množica podjetij ni velika, ampak bo zadoščala za dosego našega cilja.

Zaradi preglednosti smo imena algoritmov v Tabeli 4.1 zamenjali s kraticami. Preizkusili smo napovedovanje *številčne spremembe stopnje* in napovedovanje *spremembe stopnje v odstotkih*. V prvih štirih stolpcih so prikazani poprečno absolutno (angl. MAE) in relativno absolutno napako (angl. RAE) dveh preizkušenih variant za omrežja družbenikov, v drugih štirih pa za omrežja podjetij.

Kot lahko opazimo, so napake pri napovedovanju za podjetja precej večje v primerjavi z napakami, ki jih dobimo pri napovedovanju za družbenike.

| | Družbeniki | | | | Podjetja | | | |
|---|---|---|---|---|---|---|---|---|
| | Sprememba | | Odstotek | | Sprememba | | Odstotek | |
| | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** |
| **RF** | 5.3597 | 0.2824 | 4.8833 | 0.5414 | 13.7318 | 0.5675 | 44.6336 | 1.0465 |
| **RT** | 3.4662 | 0.1894 | 3.3087 | 0.3668 | 14.8904 | 0.6154 | 52.6351 | 1.2342 |
| **kNN** | 2.3724 | 0.1296 | 2.7642 | 0.3064 | 8.4145 | 0.3477 | 43.3021 | 1.0153 |
| **SVMR** | 10.5604 | 0.5770 | 6.14 | 0.6807 | 21.0746 | 0.8709 | 25.7451 | 0.5973 |

Tabela 4.1: Regresijsko napovedovanje številčne spremembe stopnje in spremembe stopnje v odstotkih za omrežja družbenikov in podjetij na 5-letnem časovnem oknu

Pri napovedovanju *številčne spremembe stopnje*, najmanjšo napako dobimo z algoritmom *kNN*, ki se zmoti v 34.77% vseh primerov. Število samo po sebi ni veliko in je dokaj realno, vendar če primerjamo z istim primerom pri omrežju družbenikov, bomo opazili, da je napaka 3-krat večja. Pri družbenikih se algoritem *kNN* zmoti v 12.96% vseh primerov, povprečna napaka pa znaša 2.3724.

Da bi se zares prepričali v prejšnji trditvi, smo naredili primerjavo še za drugo varianto. Pri napovedovanju spremembe v odstotkih za podjetja, najmanjšo napako dobimo z algoritmom *SVM regresija*. V tem primeru, bi se napovedni model, zgrajen po tem algoritmu, zmotil v 59.73% vseh primerov. V vseh ostalih primerih je napaka neprimerljivo večja. Že hiter pogled na isti primer za družbenike pokaže, da je tam napaka večja. Vendar, če pogledamo napake pri drugih algoritmih, je le-ta skoraj dvojno manjša. Pomembno je omeniti, da, čeprav se vsi algoritmi uporabljajo za regresijsko napovedovanje, so med seboj različni in vsi dajejo različne rezultate. V našem primeru, večina uporabljenih algoritmov pri omrežju družbenikov napovedujejo boljše v primerjavi z napovedovanjem pri omrežju podjetij. To nas pripelje do sklepa, da res ni smiselno analizirati omrežja podjetij, kadar imamo druga omrežja (omrežja družbenikov), ki so bolj primerni za analizo.

## 4.3 Vpliv časovnega okna na napovedovanje

Analizo v diplomski nalogi smo izvajali za podatke, ki so bili zajeti za več časovnih oknih. Pri zajemanju podatkov se je pojavila težava s sporadično spremembo stopnje. Poskušali smo časovnega okna povečati, kako bi spremembe stopenj postopoma skrčili. Izkazalo se je, da je bilo povečevanje časovnega okna ključnega pomena, ker se je sporadičnost zmanjšala. Ekstremne spremembe stopenj so bile predvsem opazne pri podatkih, zajetih na 2-letnem in 3-letnem časovnem oknu.

| **Napovedovanje številčne spremembe stopnje** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **2 leti** | | **3 let** | | **4 let** | | **5 let** | |
| | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** |
| **RF** | 4.3968 | 0.2070 | 7.1135 | 0.2402 | 7.3841 | 0.3243 | 5.3597 | 0.2928 |
| **RT** | 1.2961 | 0.0610 | 0.3572 | 0.0121 | 3.2758 | 0.1439 | 3.4662 | 0.1894 |
| **kNN** | 1.7229 | 0.0811 | 0.4633 | 0.0156 | 2.9581 | 0.1299 | 2.3724 | 0.1296 |
| **SVMR** | 11.7637 | 0.5539 | 16.8569 | 0.5693 | 12.7980 | 0.5620 | 10.5604 | 0.5770 |

Tabela 4.2: Regresijsko napovedovanje številčne spremembe stopnje za omrežja družbenikov na vseh časovnih oknih

V Tabeli 4.2 so vidne napake pri napovedovanju številčne spremembe stopnje na vseh časovnih oknih. Opazujmo napake, dobljene z algoritmoma *RT* in *kNN* pri 2-letnem in 3-letnem časovnem oknu. Napake so izredno majhne, kar je seveda čudno, saj ne obstaja model, ki napoveduje s 6% napako. Težava v teh primerih je ravno sporadičnost spremembe. V skoraj 75% učne množice, zajete za 2-letno časovno okno in 85% učne množice, zajete za 3-letno časovno okno se spremembe ne dogajajo. Model se tako uči na zelo malo število primerov, kjer je zares bila sprememba. Zaradi tega je treba te napake jemati z rezervo.

Če gledamo napake, dobljene pri 4-letnih in 5-letnih časovnih oknih bomo opazili, da je zgodba čisto drugačna. V teh učnih množicah se je dvojno povečalo število primerov kjer je sprememba bila, kar je modelu dalo več

učnega »materiala«. Je res, da so relativne napake večje kot prej, vendar je povsem sprejemljivo in se lahko verjame, da obstaja model, ki napoveduje s 13% in 18% napako.

| Napovedovanje spremembe stopnje v odstotkih | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 leti | | 3 let | | 4 let | | 5 let | |
| | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** | **MAE** | **RAE** |
| **RF** | 3.0385 | 0.1523 | 3.9416 | 0.1391 | 9.1227 | 0.3396 | 17.3299 | 0.4708 |
| **RT** | 0.5332 | 0.0267 | 0.3585 | 0.0127 | 10.3406 | 0.3850 | 30.4740 | 0.8278 |
| **kNN** | 0.4207 | 0.0211 | 0.4597 | 0.0162 | 6.8727 | 0.2559 | 11.8943 | 0.3231 |
| **SVMR** | 11.1055 | 0.5565 | 16.4997 | 0.5825 | 16.2856 | 0.6063 | 24.1597 | 0.6563 |

Tabela 4.3: Regresijsko napovedovanje spremembe stopnje v odstotkih za omrežja družbenikov na vseh časovnih oknih

V Tabeli 4.3 so prikazane napake pri napovedovanju spremembe stopnje v odstotkih na vseh časovnih oknih, na katerih smo preizkušali. V prvih dveh primerih (2-letno in 3-letno časovno okno), najmanjšo napako dobimo pri napovedovanju na 3-letnem časovnem oknu, ki sicer znaša 1.62%. Kot lahko opazimo, sporadičnost spremembe v tem primeru pride do še večjega izraza, saj ni realno, da bi obstajal napovedni model, ki bi se zmotil v 1% primerov. Takšen izhod je skoraj popoln, vendar to ni možno.

Pri napovedovanju na 4-letnem in 5-letnem časovnem oknu je relativna napaka zrasla. Najnižjo smo dobili pri 4-letnem časovnem oknu, znaša pa 25.59%, kar je v bistvu zelo realno.

## 4.4 Klasifikacija lastništva družbenikov

V tem poglavju bomo predstavili natančnost posameznih algoritmov pri preizkušanju klasifikacijske spremembe. Analizo smo spet izvajali na podatke, ki so bili zajeti za več časovnih obdobjih. Meritve, s katerimi bomo ocenjevali natančnost algoritmov smo opisali v Poglavju 2.3.3.5.

| | 2 leti | | | | |
|---|---|---|---|---|---|
| | **CA** | **SENS** | **SPEC** | **AUC** | **BRIER** |
| **NB** | 0.9602 | 0.9695 | 0.9560 | 0.9665 | 0.0995 |
| **kNN** | 0.9550 | 0.9661 | 0.9499 | 0.9814 | 0.0751 |
| **CT** | 0.9529 | 0.9661 | 0.9469 | 0.9701 | 0.0789 |
| **RF** | 0.9581 | 0.9695 | 0.9530 | 0.9847 | 0.0857 |

Tabela 4.4: Klasifikacijsko napovedovanje na 2-letnem časovnem oknu

V Tabeli 4.4 so predstavljene točnosti uporabljenih klasifikacijskih algoritmov, podatki pa so bili zajeti na 2-letnem časovnem oknu. Kot lahko vidimo, se izjemno dobro napovedovanje modela pri takšnem časovnem oknu nadaljuje, podobno kot pri regresijskem napovedovanju (glej Poglavje 4.3). Takšno obnašanje je bilo pričakovano, razlog za to pa smo opisali v Poglavju 4.3. Osredotočimo se na zadnjem stolpcu oz. na Brier oceno. Pri Brier oceni, vrednost 0 pomeni najboljša možna ocena, vrednost 1 pa najslabša možna. V našem primeru smo pri vseh algoritmih dobili oceno, ki je zelo blizu najboljšemu možnemu izhodu. Poudarjamo, da je velikost razreda 1 neprimerljivo manjša v primerjavi z razreda 0. Ni možno, da je napovedovanje toliko dobro, če je število primerov, na katerih se model uči, izredno majhno.

| | 3 leta | | | | |
|---|---|---|---|---|---|
| | **CA** | **SENS** | **SPEC** | **AUC** | **BRIER** |
| **NB** | 0.9827 | 0.9060 | 1 | 0.9799 | 0.0350 |
| **kNN** | 0.9874 | 0.9487 | 0.9961 | 0.9821 | 0.0242 |
| **CT** | 0.9905 | 0.9573 | 0.9981 | 0.9798 | 0.0190 |
| **RF** | 0.9811 | 0.9060 | 0.9981 | 0.9925 | 0.0444 |

Tabela 4.5: Klasifikacijsko napovedovanje na 3-letnem časovnem oknu

V Tabeli 4.5 so predstavljene točnosti uporabljenih klasifikacijskih algoritmov, kjer so podatki bili zajeti na 3-letnem časovnem oknu. V tem primeru algoritmi delajo še boljše kot prej, saj je pri vsakem preizkušenem algoritmu

Brier ocena skoraj trikrat nižja. Tudi v prvem stolpcu, kjer je predstavljena klasifikacijska točnost posameznega algoritma, so vrednosti zelo blizu 1, kar samo potrjuje, da je napovedovanje izredno dobro. Enako je bilo tudi pri regresijskem napovedovanju (glej Tabelo 4.2 in Tabelo 4.3), kjer je bila napaka napovednega modela najnižja. Še enkrat poudarjamo, da je treba te napake jemati z rezervo.

| | **4 let** | | | | |
|---|---|---|---|---|---|
| | **CA** | **SENS** | **SPEC** | **AUC** | **BRIER** |
| **NB** | 0.8054 | 0.9212 | 0.6693 | 0.8814 | 0.2830 |
| **kNN** | 0.9171 | 0.9257 | 0.9089 | 0.9642 | 0.1351 |
| **CT** | 0.8339 | 0.7898 | 0.8752 | 0.9017 | 0.2426 |
| **RF** | 0.8688 | 0.8153 | 0.9188 | 0.9664 | 0.1777 |

Tabela 4.6: Klasifikacijsko napovedovanje na 4-letnem časovnem oknu

V Tabeli 4.6 so prikazane točnosti uporabljenih klasifikacijskih algoritmov, zagnanih na podatke za 4-letnem časovnem oknu. Kot lahko vidimo, vrednosti prikazanih metrik se znižujejo, v skladu z našimi pričakovanji. Pri algoritmu *kNN* so vrednosti metrik še vedno visoke, vendar pri ostalih algoritmih so za vsaj 10% nižje, kar nam daje boljšo sliko o tem, kateri algoritmi so bolj zanesljivi.

| | **5 let** | | | | |
|---|---|---|---|---|---|
| | **CA** | **SENS** | **SPEC** | **AUC** | **BRIER** |
| **NB** | 0.7274 | 0.6567 | 0.7765 | 0.7723 | 0.3670 |
| **kNN** | 0.9719 | 0.9552 | 0.9847 | 0.9809 | 0.0495 |
| **CT** | 0.8204 | 0.7512 | 0.8736 | 0.8956 | 0.2417 |
| **RF** | 0.7403 | 0.4046 | 0.9987 | 0.9758 | 0.2191 |

Tabela 4.7: Klasifikacijsko napovedovanje na 5-letnem časovnem oknu

V Tabeli 4.7 se lahko opazijo točnosti uporabljenih klasifikacijskih algoritmov, algoritme pa smo testirali na podatke za 5-letnem časovnem oknu.

Pri testiranju, smo se osredotočili na napovedovanje razreda 1, ker je takšnih primerov v podatkovni množici manj. Tako kot pri regresijskem napovedovanju, so tudi pri klasifikacijskem napovedovanju meritve sprejemljive, saj lahko verjamemo, da obstaja napovedni model, ki se zmoti v 20-25% vseh primerov.

Če izvzamemo algoritma *kNN*, kjer so meritve še vedno visoke, najbolj sprejemljive rezultate daje algoritem *klasifikacijsko drevo*, ki je v tabeli zaradi preglednosti označen z okrajšavo *CT*. Na splošno, natančnost algoritma znaša 82%, kar je razvidno iz atributa *klasifikacijska natančnost*. Drugi atribut v Tabeli 4.7 je *občutljivost*. *Občutljivost* v tem primeru znaša 75.12%, kar pomeni, da smo pravilno napovedali spremembo 75.12% od vseh vozlišč v razredu 1. *Specifičnost* se uporablja skupaj z *občutljivostjo* in meri ravno obratno oz. odstotek tistih vozlišč v razredu 0, katerim nismo napovedali spremembo in katerim se ni zgodila sprememba. V našem primeru, ta znaša 87.36%.

## 4.5 Luščenje in uporaba pridobljenega znanja

Iz vseh opravljenih analiz smo na koncu dobili napovedni model, ki na podlagi vnaprej določenih pravil v klasifikacijskem drevesu, napoveduje, ali se bo zgodila sprememba v stopnji določenega vozlišča.

V Poglavju 4.1 smo razlagali o izbranih atributih, ki najbolj vplivajo na natančnost napovedovanja. To so *središčnost lastnega vektorja, stopnja vozlišča in ekscentrična središčnost*. Kot primer, na katerega bomo razložili uporabo klasifikacijskega drevesa na Sliki 4.1, recimo da imamo vozlišče, ki ima *stopnjo* 85, *središčnost lastnega vektorja* naj znaša 0.120, *ekscentrična središčnost* pa naj bo 5. Bistvo klasifikacijskega drevesa je, da na podlagi predpisanih pravil, pridemo do listov drevesa, ki nam v bistvu povejo, kakšen bo rezultat napovedovanja.

Najprej, pogledamo koren drevesa, oz. pravilo na vrhu in vrednosti dejanskega vozlišča primerjamo z vrednostmi v pravilih. To delamo, tako kot

**Tree**

| Classification Tree | Class | P(Class) | P(Target) | # Inst | Distribution (rel) | Distribution (abs) |
|---|---|---|---|---|---|---|
| | 0 | 0.565 | 0.435 | 1386 | 0.565:0.435 | 783:603 |
| eccen_cent <=9.500 | 1 | 0.684 | 0.684 | 519 | 0.316:0.684 | 164:355 |
| eigen_cent <=0.010 | 1 | 0.591 | 0.591 | 235 | 0.409:0.591 | 96:139 |
| eigen_cent <=0.007 | 1 | 0.714 | 0.714 | 175 | 0.286:0.714 | 50:125 |
| eccen_cent <=7.500 | 1 | 0.712 | 0.712 | 153 | 0.288:0.712 | 44:109 |
| eccen_cent >7.500 | 1 | 0.727 | 0.727 | 22 | 0.273:0.727 | 6:16 |
| eccen_cent <=8.500 | 0 | 0.857 | 0.143 | 7 | 0.857:0.143 | 6:1 |
| eccen_cent >8.500 | 1 | 1.000 | 1.000 | 15 | 0.000:1.000 | 0:15 |
| eigen_cent >0.007 | 0 | 0.767 | 0.233 | 60 | 0.767:0.233 | 46:14 |
| eccen_cent <=8.000 | 0 | 0.958 | 0.042 | 48 | 0.958:0.042 | 46:2 |
| eccen_cent >8.000 | 1 | 1.000 | 1.000 | 12 | 0.000:1.000 | 0:12 |
| eigen_cent >0.010 | 1 | 0.761 | 0.761 | 284 | 0.239:0.761 | 68:216 |
| eigen_cent <=0.117 | 1 | 0.611 | 0.611 | 175 | 0.389:0.611 | 68:107 |
| degree <=22.000 | 1 | 0.800 | 0.800 | 25 | 0.200:0.800 | 5:20 |
| eccen_cent <=6.000 | 1 | 1.000 | 1.000 | 20 | 0.000:1.000 | 0:20 |
| eccen_cent >6.000 | 0 | 1.000 | 0.000 | 5 | 1.000:0.000 | 5:0 |
| degree >22.000 | 1 | 0.580 | 0.580 | 150 | 0.420:0.580 | 63:87 |
| eccen_cent <=5.500 | 0 | 0.521 | 0.479 | 121 | 0.521:0.479 | 63:58 |
| eccen_cent >5.500 | 1 | 1.000 | 1.000 | 29 | 0.000:1.000 | 0:29 |
| eigen_cent >0.117 | 1 | 1.000 | 1.000 | 109 | 0.000:1.000 | 0:109 |
| eccen_cent >9.500 | 0 | 0.714 | 0.286 | 867 | 0.714:0.286 | 619:248 |
| degree <=211.500 | 0 | 0.723 | 0.277 | 855 | 0.723:0.277 | 618:237 |
| eigen_cent <=0.028 | 0 | 0.637 | 0.363 | 653 | 0.637:0.363 | 416:237 |
| eccen_cent <=10.500 | 0 | 0.837 | 0.163 | 282 | 0.837:0.163 | 236:46 |
| eccen_cent >10.500 | 1 | 0.515 | 0.515 | 371 | 0.485:0.515 | 180:191 |
| eigen_cent <=0.011 | 0 | 0.604 | 0.396 | 298 | 0.604:0.396 | 180:118 |
| eigen_cent >0.011 | 1 | 1.000 | 1.000 | 73 | 0.000:1.000 | 0:73 |
| eigen_cent >0.028 | 0 | 1.000 | 0.000 | 202 | 1.000:0.000 | 202:0 |
| degree >211.500 | 1 | 0.917 | 0.917 | 12 | 0.083:0.917 | 1:11 |

Slika 4.1: Klasifikacijsko drevo, rezultat napovedovanja na 5-letnem časovnem oknu

smo prej omenili, dokler ne pridemo do listov drevesa. V našem primeru smo za naše vozlišče napovedali spremembo, kar pomeni, da bo po preteku 5. let, vozlišče imelo višjo stopnjo kot jo ima sedaj.

# Poglavje 5

# Sklepne ugotovitve

V diplomski nalogi smo z uporabo različnih metod strojnega učenja, podatkovnega rudarjenja in statistike, na osnovi analiziranja družbenih omrežij, poskušali napovedovati kako se v omrežju družbenikov spreminja lastništvo. V našem primeru je omrežje zasnovano na podatkih, ki se lahko najdejo na spletni strani AJPES-a. Bolj natančno, iz vseh podatkov smo izluščili podatke o poslovnih deležih družbenikov in njim ustreznih podjetij, da bi lahko napovedovali lastništvo. Sprememba lastništva v omrežju je v bistvu sprememba stopnje določenega vozlišča.

Naslednji korak je bil računanje različnih metrik, ki odražajo pomembnost vozlišča v omrežju. Z uporabo programskega paketa Orange smo tako oblikovane podatke analizirali. Pri tem smo uporabili različne algoritme in metode podatkovnega rudarjenja, da bi ugotovili, kateri algoritem daje najboljše rezultate.

Končni rezultat celotne analize je napovedni model, ki se lahko zanesljivo uporablja pri napovedovanju stopnje vozlišč v omrežju. Ni nujno, da stopnja vozlišč predstavlja zgolj lastništvo, tako kot v našem primeru. Model je lahko uporaben tudi na drugih področjih.

Ljudje smo po naravi takšni, da česarkoli se lotimo, poskušamo na vsak način to stvar narediti čim boljše. Enako velja tudi za to analizo, ki se lahko s par spremembami v podatkih in v kodi izboljša.

Značilnost AJPES-a je, da lahko dobimo podatke o podjetjih ali družbenikih, ki so že zdavnaj bankrotirali, šli v stečaj, so bili likvidirani, itn. To pomeni, da se podatki o takšnih poslovnih osebkih ne brišejo iz njihove baze, ampak se ustrezno označijo. Razlog, zaradi katerega je določena poslovna enota prenehala s poslovanjem, se dopiše zraven imena poslovne enote in se potem v bazi ustrezno označi. Tako je možno, da na AJPES-ovi spletni strani lahko iščemo po aktivnih in izbrisanih enotah. Za našo analizo to pomeni, da bi lahko bile prisotne še negativne spremembe stopnje vozlišč. Negativno spremembo stopnje lahko uporabimo na več načinov.

En primer uporabe je, da bi klasifikacijskemu napovedovanju dodali še en razred, s čimer bi skupno imeli tri razrede. Razreda 0 in 1 bi se uporabljala isto kot prej, razred -1 pa bi se uporabljal za napovedovanje negativne spremembe stopnje.

Drugi primer uporabe je napovedovanje ali bo oseba bankrotirala. Recimo, da za določeno vozlišče v naslednjem obdobju napovemo negativno stopnjo. Če je absolutna vrednost napovedane stopnje večja kot stopnje v trenutku napovedi, se napove stečaj poslovne enote.

Za obe omenjeni nadgradnji bi bila potrebna predelava uporabljenih podatkov, da bi izbrisali vozlišča, ki ustrezajo poslovnim enotam, ki so prenehale z delovanjem. Ker nam trenutna ontologija ne ponuja možnosti zajemanja takšnih podatkov, bi bilo treba spremeniti ontologijo, v skladu z organizacijo zapisov na AJPES-ovi spletni strani. Podatki o organizacijskih spremembah so vidni v zavihku Skupščinski sklepi (glej Sliko 3.1). Poleg tega, potrebno bi bilo pridobiti dovoljenje s strani AJPES-a, ker njihovi pogoji uporabe omogočajo uporabo le neznatnega dela njihove podatkovne baze.

Tretji način nadgradnje obstoječe analize je, da bi uporabili pristop, kjer bi dejansko napovedovali ustvarjanje novih povezav. *Napovedovanje povezav* (angl. link prediction)[29] je področje, s katerim se ukvarja vse več in več strokovnjakov. Obstaja več načinov, kako izvesti napovedovanje povezav, ki sicer temelji na metrike, ki povzemajo celotno strukturo grafa in neposredno vplivajo na ustvarjanje novih povezav v omrežju. Primere takšnih metrik so

*koeficient nakopičenosti* (angl. Clustering coefficient), *povprečna dolžina poti* (angl. Average path length) in *stopnja*.

# Literatura


[1] Natalya F. Noy, Deborah L. McGuinness
*Ontology Development 101: A Guide to Creating Your First Ontology.*.
Dostopno na:
`http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf`

[2] Web ontology language, uradna spletna stran. Dostopno na:
`http://www.w3.org/OWL/`

[3] Resource description framework, uradna spletna stran. Dostopno na:
`http://www.w3.org/RDF/`

[4] Opsahl, T., 2013.
*Triadic closure in two-mode networks: Redefining the global and local clustering coefficients.* Social Networks 35, doi:10.1016/j.socnet.2011.07.001. Dostopno na:
`http://toreopsahl.com/tnet/two-mode-networks/defining-two-mode-networks/`

[5] M. E. J. Newman
*Networks: an introduction*, Chapter 7: Measures and metrics

[6] Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, Wei Wang
*Data Mining Curriculum: A Proposal.* Dostopno na:
`http://www.cs.uiuc.edu/~hanj/kdd_curriculum.pdf`

[7] Fisher R.A.
*Annals of Eugenics 7*, The use of multiple measurements in taxonomic problems, 179-188

[8] Alan O. Sykes
*An Introduction to Regression Analysis*. Dostopno na:
`http://www.law.uchicago.edu/files/files/20.Sykes_.Regression.pdf`

[9] Harry Zhang
*The Optimality of Naive Bayes*. Dostopno na:
`http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf`

[10] K. Ming Leung
*k-Nearest Neighbor Algorithm for Classification*. Dostopno na:
`http://cis.poly.edu/~mleung/FRE7851/f07/k-NearestNeighbor.pdf`

[11] Roman Timofeev
Classification and Regression Trees (CART): Theory and Applications. Dostopno na:
`http://tigger.uic.edu/~georgek/HomePage/Nonparametrics/timofeev.pdf`

[12] Leo Breiman
*RANDOM FORESTS*. Dostopno na:
`http://oz.berkeley.edu/~breiman/randomforest2001.pdf`

[13] Andy Liaw and Matthew Wiener
*Classification and Regression by randomForest*. Dostopno na:
`http://cogns.northwestern.edu/cbmg/LiawAndWiener2002.pdf`

[14] Zizhen Yao and Walter L. Ruzzo
*A Regression-based K nearest neighbor algorithm for gene function pre-*

*diction from heterogeneous data.* Dostopno na:
`http://www.biomedcentral.com/1471-2105/7/S1/S11`

[15] Alex J. Smola and Bernhard Scholkopf
*A Tutorial on Support Vector Regression.* Dostopno na:
`http://alex.smola.org/papers/2003/SmoSch03b.pdf`

[16] B.Azhagusundari, Antony Selvadoss Thanamani
*Feature Selection based on Information Gain.* Dostopno na:
`http://www.ijitee.org/attachments/File/v2i2/B0352012213.pdf`

[17] Sanasam Ranbir Singh, Hema A. Murthy, Timothy A. Gonsalves
*Feature Selection for Text Classification Based on Gini Coefficient of Inequality.* Dostopno na:
`http://jmlr.org/proceedings/papers/v10/sanasam10a/sanasam10a.pdf`

[18] E. L. Lehman, George Casella
*Theory of Point Estimation, 2nd ed.*

[19] Marko Robnik Šikonja, Igor Kononenko
*Theoretical and Empyrical analysis of ReliefF and RReliefF.* Dostopno na:
`http://lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf`

[20] Rob J Hyndman
*Another look at measures of forecast accuracy.* Dostopno na:
`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.9771&rep=rep1&type=pdf`

[21] Igor Kononenko
*Strojno učenje, stran 75*

[22] NCSSM Statistics Leadership Institute
*Sensitivity and Specificity.* Dostopno na:

http://courses.ncssm.edu/math/Stat_Inst/Stats2007/Stat%20and%20Calc/Sensitivity%20and%20Specificity.pdf

[23] Tom Fawcett
*An introduction to ROC analysis*. Dostopno na:
http://people.inf.elte.hu/kiss/12dwhdm/roc.pdf

[24] Tom Fawcett
*ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. Dostopno na:
http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf

[25] Beth Ebert
*Methods and scores used for verifying ensemble forecasts*. Dostopno na:
http://www.cawcr.gov.au/projects/EPSverif/scores/scores.html

[26] Dokumentacija OWL programskega vmesnika. Dostopno na:
http://owlapi.sourceforge.net/

[27] Dokumentacija in primeri, narejeni s pomočjo Gephi programskega vmesnika. Dostopno na:
https://gephi.org/toolkit/

[28] Dokumentacija programskega paketa Orange. Dostopno na:
http://orange.biolab.si/docs/latest/

[29] The link prediction problem for social networks. Dostopno na:
http://www.cs.cornell.edu/home/kleinber/link-pred.pdf