Sudbina životinja u skloništima

Iva Hršak, Ana Iveković i Ana Mlinarić

***Sažetak*— U ovom radu opisali smo naš pristup rješavanju problema predviđanja ishoda sudbina životinja u skloništima. Ovaj problem predstavljen je u natjecanju 'Shelter Animal Outcomes" na Kaggle-u. U rješavanju problema primijenili smo tri modela strojnog učenja SVM (Support Vector Machines), Random Forest I XGBoost (Extreme Gradient Boosting).**

***Ključne riječi*— SVM, Random Forest, XGBoost, kaggle.**

I. Uvod

Svake godine u SAD-u otprilike 7.6 milijuna kućnih ljubimaca završi u skloništima za životinje. Dio životinja nađe novi dom kod udomitelja, no mnogi nisu te sreće. Potresna je informacija da je u SAD-u svake godine eutanizirano 2.7 milijuna pasa i mačaka. Cilj našeg projekta je na temelju dobivenih podataka predvidjeti kakav će biti ishod za pojedinu životinju pa na temelju tih podataka poduzeti neke dodatne mjere kako bi se životinje s manjom vjerojatnošću udomljavanja ipak udomile.

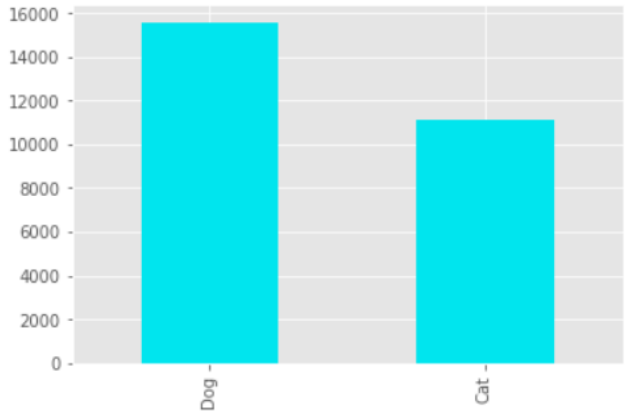
II. PODACI

Skup podataka sastoji se od informacija o gotovo 27 000 životinja. Preuzeti su od američkog skloništa za životinje Austin Animal Center, a prikupljani su od listopada 2013. godine do ožujka 2016. godine. Za svaku je životinju dan njezin ID, ime, datum unosa podataka, tip, vrsta, spol, starost, boja, informacija o kastraciji ili sterilizaciji te ishod, koji može biti jedno od sljedećeg: udomljavanje, smrt, eutanazija, povratak vlasniku ili prijenos u drugo sklonište.

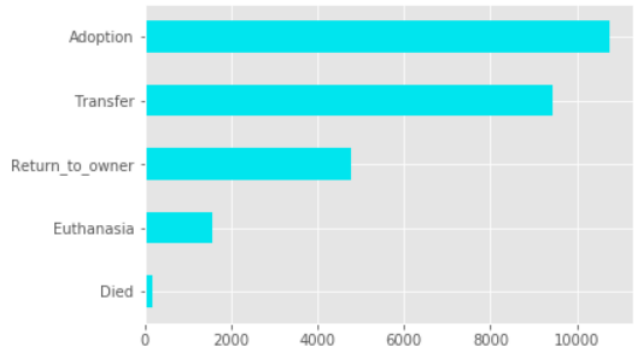
A. Analiza podataka

Izvršili smo neke jednostavnije analize skupa za učenje kako bismo vidjeli kakvim podacima raspolažemo i eventualno uočili neka pravila ili međusobne ovisnosti svojstava i ishoda. Također, na skupu primjera za učenje provjerili smo neke tvrdnje koje smatramo zanimljivima i korisnima za daljnje istraživanje.

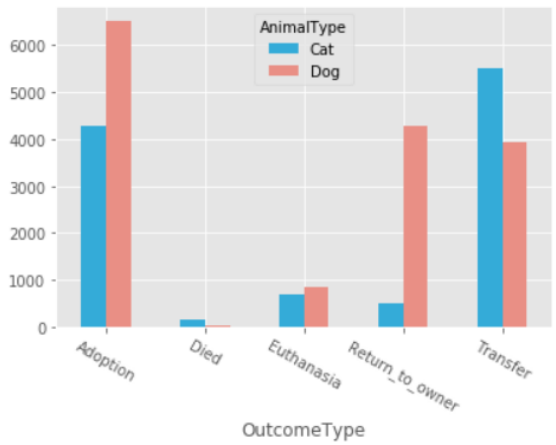
Prvo promatramo vrstu životinje, odnosno radi li se o psu ili mački. Obradom podataka dobiva se da ima 58.3% pasa i 41.7% mačaka.



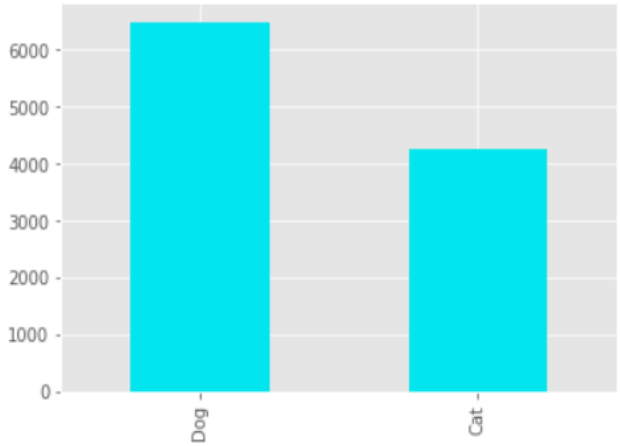
Sljedeća bitna informacija nam je pripadnost životinja određenoj klasi, odnosno informacija o tome je li životinja udomljena, eutanizirana, vraćena vlasniku, umrla ili je prenesena u drugo sklonište za životinje.



Sada spajamo ove dvije informacije, tj. Prikazujemo podjelu pasa i mačaka u pet navedenih klasa.



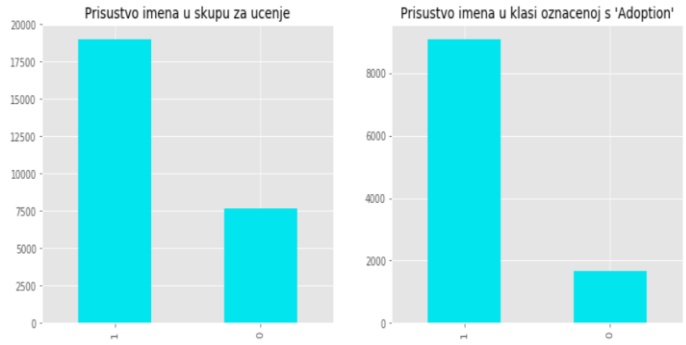
Analizom dobivamo podatak da je posvojeno 41.7% pasa i 38.4% mačaka
Promotrimo sada detaljnije klasu 'Adoption'.



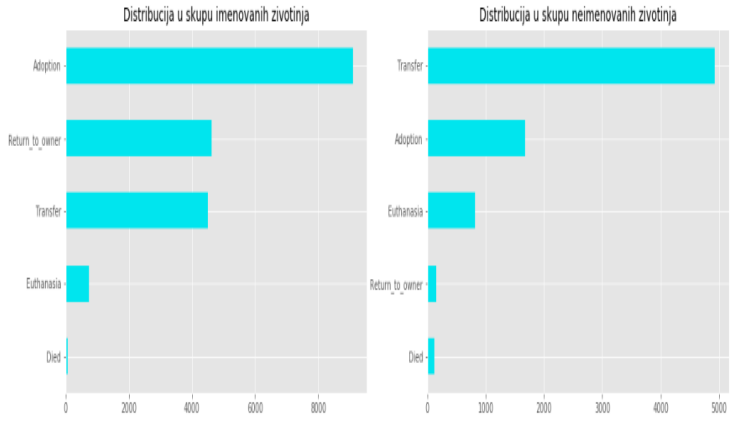
Po grafovima se čini da je postotak mačaka i pasa sličan onome u početnom skupu. Provjerili smo to na podacima, te dobivamo da je u skupu klasificiranom s 'Adoption' 60.3% pasa i 39.7% mačaka. Jedna je od naših tvrdnji bila da je vjerojatnost udomljavanja za pse i mačke podjednaka, što smatramo potvrđenim na danom skupu za učenje.

Sljedeće što nas zanima jest možemo li davanjem imena životinji povećati vjerojatnost njezinog pozitivnog ishoda. Inače ćemo pozitivnim ishodom smatrati udomljavanje i povratak vlasniku, no u ovom ćemo slučaju zasebno provjeravati svaku klasu. Naime, smatramo da, ukoliko je životinja već imala i vlasnika i vlasnik joj je dao ime, ono ime koje je dobila u skloništu neće utjecati na njezin povratak vlasniku.

Sada promatramo koliko životinja u početnom skupu, odnosno u klasi 'Adoption' ima ime te uspoređujemo te dvije informacije. Na grafu je prisustvo imena označeno s '1', a odsustvo s '0'.



Prema gornjim grafovima, mogli bismo reći da bi davanje imena životinji doista moglo poboljšati njezine šanse za udomljavanje. No, pogledajmo još kakva je točno distribucija po klasama za imenovane i neimenovane životinje.

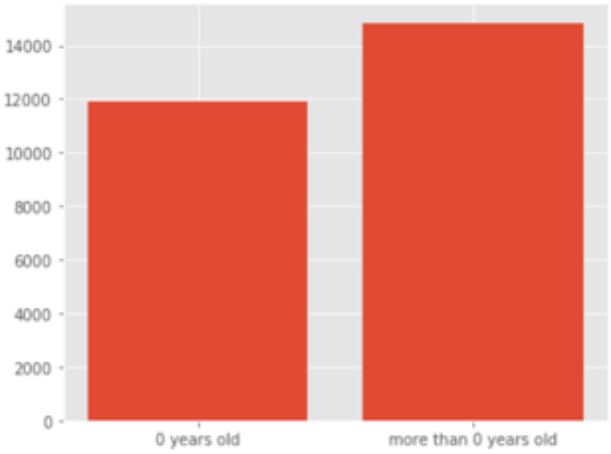


Vidimo da je poredak ishoda pogodniji za životinje koje imaju ime. Ima smisla da je najveći broj neimenovanih životinja prebačen u drugo sklonište. Tamo će onda dobiti ime. Ipak, vidimo da ima životinja bez imena koje jesu udomljene pa možda ne bi bilo loše da, dok čekaju svoj transfer, dobiju neko ime za bolju reklamu. Veće su šanse da im možda uopće neće trebati transfer. Što se tiče eutanazije, ona uzima više maha u skupu neimenovanih životinja. To bi moglo biti i zato što su to vjerojatno tek nađene životinje, od kojih su mnoge ranjene ili neprilagođene pa ih se eutanazira prije nego što su imale priliku dobiti ime.

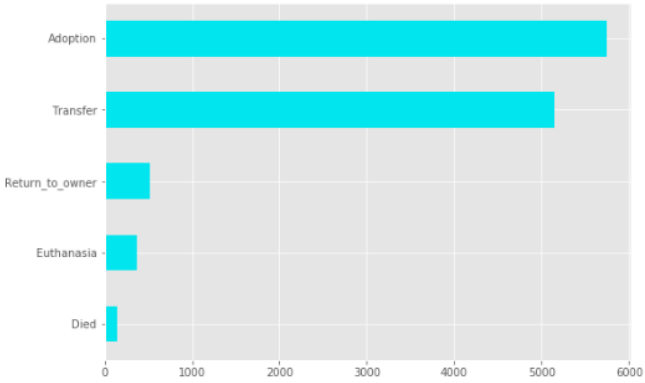
Nadalje, promatramo starost životinje. Jedna od hipoteza koje smo naveli u prijedlogu projektne teme jest da je za starije životinje veća vjerojatnost povratka vlasniku jer je vjerojatno povezanost između njih veća. U ovom ćemo dijelu istražiti općenito kako starost životinje utječe na ishode i provjeriti što podatci kažu o našoj hipotezi.

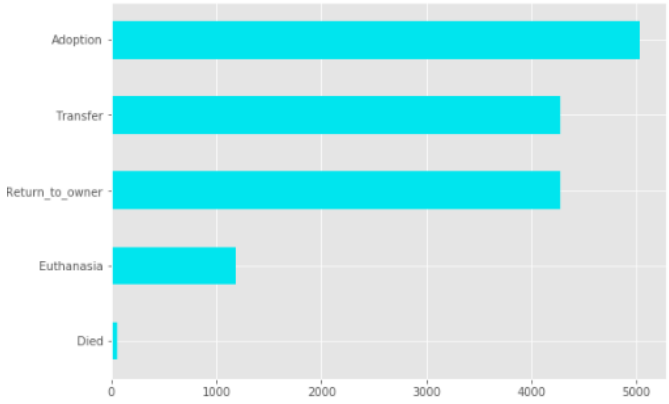
U očišćenom skupu podataka podatci o starosti su u danima radi preciznijeg učenja modela za klasifikaciju.

Ovdje ćemo, radi jednostavnosti, koristiti podatke u godinama.
Starost životinja kreće se od 0 do 20 godina. Pogledajmo kako su početni podaci podijeljeni s obzirom na starost.

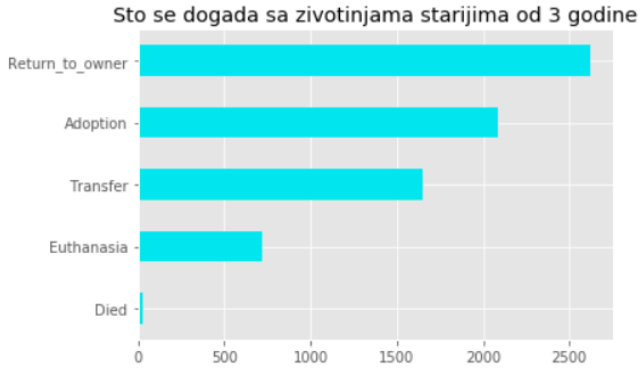


Vidimo da je najzastupljenija skupina štenaca i mačića mlađih od godinu dana. Iz gornjih grafova zaključujemo da ima smisla početni skup podijeliti na životinjice starosti do godinu dana i na ostatak. Prikazujemo distribucije tih dviju skupina.

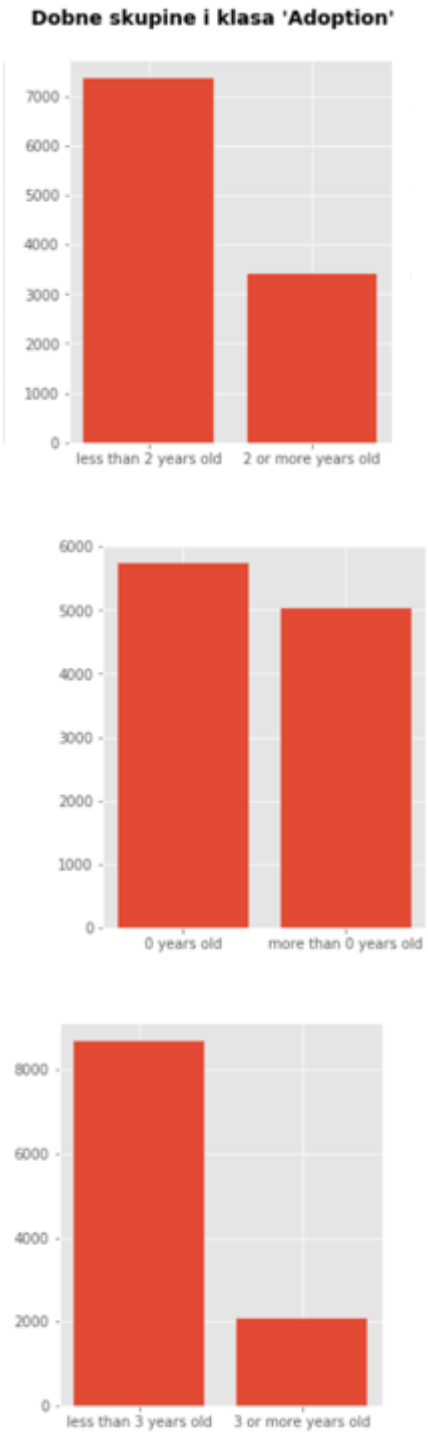




Na gornjim je grafovima očito da je vjerojatnost povratka vlasniku doista veća za životinje starije od godinu dana. Provjerimo još što se "starijoj" skupini događa ako izuzmemo vrlo mlade pse i mačke, tj. ako uzmemo životinje starosti npr. 3 godine i više.



Ovime smatramo našu hipotezu da je vjerojatnost povratka starije životinje vlasniku veća potvrđenom. Dobiveni su rezultati očekivani, ali i poučni. Govore nam kako već životinje koje napune dvije godine imaju znatno manju šansu udomljavanja, dakle već se njima treba posvetiti više pažnje. Smatramo da je to korisna informacija. Ljudi više vole udomljavati mlađe životinje, ali i životinje stare 2 ili 3 godine još su uvijek mlade. Mišljenja smo da bi se broj udomljavanja u tim skupinama mogao povećati nekim akcijama upoznavanja, reklamama i slično.



III. MODELI STROJNOG UČENJA

A. *Priprema podataka*

Podatke je potrebno pretvoriti u oblik pogodan za učenje modela. Također, potrebno je izbaciti ili dodati attribute koji bi nam pomogli kod učenja. Opisat ćemo neke od promjena koje smo mi učinili nad podacima.

Za svaku životinju dana nam je starost životinje, vrijednost je izražena u godinama, mjesecima, tjednima ili danima. Odlučili smo da ćemo sve vrijednosti izraziti u danima.

Atribut SexuponOutcome sadrži informacije o spolu životinje, ali i o tome je li životinja sterilizirana ili kastrirana.Smatramo da bi svaka od tih informacija bila važna za treniranje modela pa smo ih razdvojili u dva zasebna atributa..

Svojstvo spola pripremamo za učenje modela tako što vrijednostima pridružujemo odgovarajuće vektore. Vektor (1, 0) označava mužjaka, vektor (0, 1) ženku, a vektor (0, 0) nepoznat podatak. Vektore realiziramo dodavanjem novih stupaca, po jedan stupac za svaku komponentu.

Odlučili smo da nam samo ime životinje nije važno, da nam je važna samo činjenica ima li životinja ime ili ne.

Kako bismo mogli uključiti vrstu životinje u model, vektorom (1, 0) ili (0, 1) označit ćemo je li ona pas ili mačka.

Kako bismo mogli istražiti postoji li neki trend u udomljavanju životinja koji je povezan s datumima, godišnjim dobima i sl., razdvojili smo vremensku oznaku na dan, mjesec i godinu. Dodajemo podjelu na dane u tjednu, doba dana i godišnje doba.

Za boje smo u početnim podacima imali 366 različitih vrijednosti, uočili smo nekoliko ključnih riječi i svojstvo boje odlučili podijeliti na vektor ("white", "black", "brown", "blue", "tan", "tabby", "red", "calico", "orange", "chocolate", "gray", "tortie", "tricolor").

Broj pasmina u početku je bio 1380. Sve vrste pasa "razbijamo" u osnovne vrste. Npr. Terrier i Terrier Mix je brojan samo pod Terrier, a Terrier/Puddle Mix je brojan kao Terrier i kao Puddle. Uočili smo učestalo pojavljivanje nekih pasmina, ali na malo drugačije načine. Tako se npr. Terrier pojavljuje u Silky Terrier, Welsh Terrier, Soft Coated Wheaten Terrier itd. Takve smo slučajeve odlučili svrstati sve u istu skupinu (Terr, Retriever, Bulldog). Ako uzmemo 20 najčešćih skupina, 82% ovako podjeljenih podataka bit će obuhvaćeno tim skupinama, one koje ne možemo svrstati u ove skupine svrstali smo u skupinu Others, a one koje su svrstane u više skupina svrstat ćemo I u skupinu Mix, koja označava da je životinja mješanac.

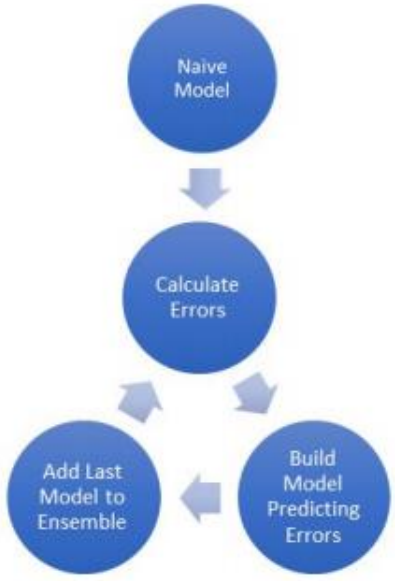
Svakom smo ishodu dodijelile broj od 0 do 4. Kako ovdje taj broj predstavlja samo krajnji ishod, nećemo te podatke koristiti za neko dodatno računanje, a u računanju greške važno je samo jesmo li ili nismo pogodili klasu, nismo imali potrebu svrstavati ih u vektor.

B. *Extreme Gradient Boosting (XGBoost)*

XGBoost (puni naziv Extreme Gradient Boosting) predstavlja brz i efikasan model za rad s tabličnim

podacima. Koristi se za nadzirano učenje i multi-class klasifikaciju. Budući da se od nas traži klasifikacija velikog broja podataka koji su u tekstualnom formatu, smatramo da bi XGBoost trebao dati dobre rezultate. XGBoost je implementacija Gradient Boosted Decision Trees algoritma, koji se može prikazati sljedećim dijagramom:.

Sastoji se od ciklusa koji grade nove modele i kombiniraju ih u već postojeći model. Potrebne su nam neke osnovne predikcije za pokretanje ciklusa pa je polazna točka naivan model koji je dan na ulazu. U praksi se pokazalo da početno predviđanje ne mora biti potpuno točno, jer će naknadne nadopune modela rješavati pogreške. Ciklus se započinje računanjem pogrešaka za svako promatranje u skupu podataka. Zatim se izrađuje novi model koji će predviđati te pogreške. Taj model se zatim dodaje u skup dosadašnjih modela. Da bismo napravili predikciju, potrebno je koristiti predikcije iz svih do sad formiranih modela. Pomoću tih predikcija, možemo izračunati nove pogreške, izraditi sljedeći model i dodati ga prethodnim modelima



Algoritam smo prvo pokrenuli s početnim ne optimalnim parametrima.

```
xgb_param_dist = {"n_estimators"     : 200,
                  "max_depth"        : 10,
                  "learning_rate"    : 0.6,
                  "colsample_bytree" : 0.7,
                  "objective"        : "multi:softmax",
                  "num_class" : 5}
```

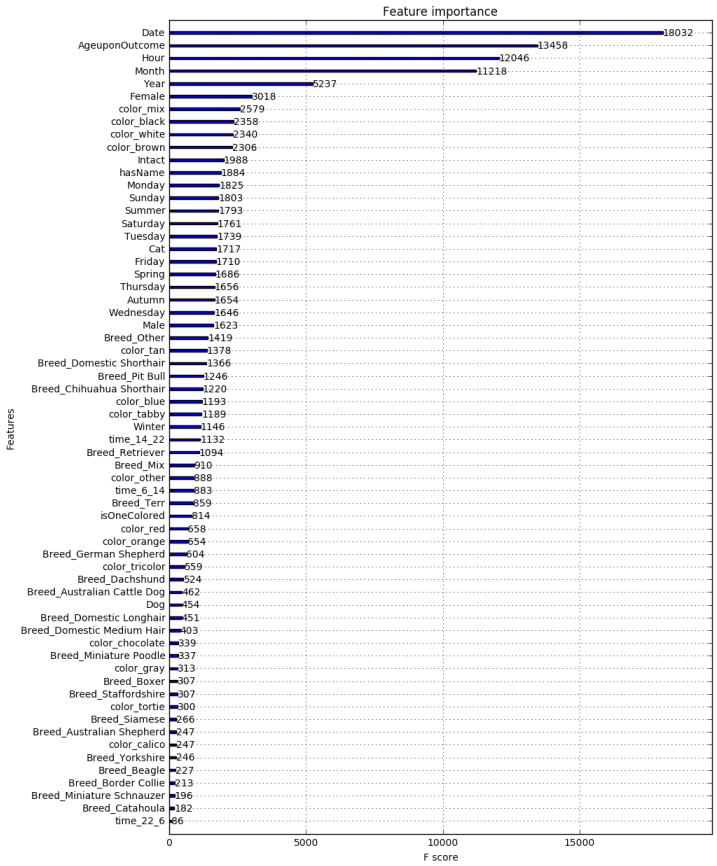
Rezultat postignut na kaggle-u s ovim parametrim je bio 1.09483.

Nakon toga tražili smo optimale hiperparametre pomoću RandomSearchCV. Dobiveni parametri su sljedeći

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=0.66763483792797729, gamma=0,
       learning_rate=0.07352705549149606, max_delta_step=0, max_depth=9,
       min_child_weight=1, missing=None, n_estimators=180, n_jobs=1,
       nthread=None, objective='multi:softprob', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=True, subsample=0.3827033431731755)
```

Nakon toga pokrenuli smo najbolji model I na kaggle-u postigli rezultat 0.72464.

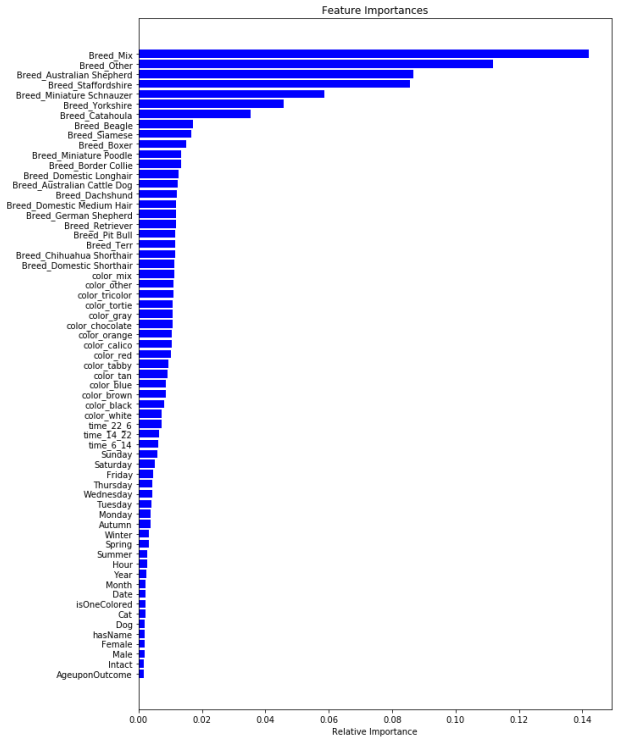
Kao rezultat treniranja modela, dobisli smo popis atributa I njihovih važnosti.



C. *Random Forest*

Random Forest algoritam za *multi-class* klasifikaciju se temelji na stablima odluke (*decision trees*). Stablo odluke klasificira objekt pitajući *da-ne* , tj. *0 - 1 pitanja*. Dobro formirano stablo svakim će pitanjem prepoloviti broj opcija, čime i za veliki broj opcija vrlo brzo dolazimo do odluke. Međutim, kod stabala odlučivanja već i na malim dubinama (npr. 5) dolazi do problema *overfitting* -a i dva stabla odlučivanja mogu davati međusobno vrlo inkonzistentne odluke. Pokazuje se da će davati konzistentne odluke na onim područjima u podacima koja nisu obuhvaćena *overfittingom* , a razlikovat će se u onima koja jesu. Dakle, ukoliko imamo informacije od više stabala s problemom *overfitting*- a, kombiniranjem njihovih odluka možemo zapravo riješiti taj

problem. *Random Forest* metoda čini upravo to.
Python biblioteka *scikit-klearn* sadrži sve potrebno za provođenje opisane metode.
Informacije o načinu rada, instalaciji i korištenju te sliku stabla odlučivanja pronašli smo na [9] .
Metodu Random Forest primjenjujemo na pripremljenim podacima. Nakon što smo istrenirali model na skupu za treniranje, radimo predikciju na testnom skupu. Predikcija ovim modelom na kaggleu je postigla rezultat 0.76717.
Prikazujemo i graf iz kojeg je vidljivo koje svojstvo ima koliku važnost.



D. Support Vector Machines
SVM je moguće pokrenuti koristeći python biblioteku sklearn. Postoji više verzija, a mi smo pokrenuli multiklasifikaciju s LinearSVC algoritmom i sa SVC algoritmom. Najbolji rezultat dobili smo sa SVC algoritmom kada smo postavili parametar C na vrijednost 5.0. Međutim, krajnji rezultat nikako nije zadovoljavajuć. Najbolje što smo na ovaj način dobili jest score=11.94204.

IV. Zaključak

U dosadašnjim istraživanjima najbolje je rezultate davala metoda XGBoost. I naš rad to potvrđuje. Smatramo da bi rezultati mogli biti bolji uz precizniju transformaciju svojstava te optimizaciju odabira svojstava.

Reference

[1] https://www.kaggle.com/c/shelter-animal-outcomes

[2] https://andraszsom.wordpress.com/2016/07/27/kaggle-for-the-paws/

[3] https://mark-borg.github.io/blog/2016/shelter-animal-competition/

[4] https://www.kaggle.com/mrisdal/quick-dirty-randomforest

[5] https://www.kaggle.com/dansbecker/learning-to-use-xgboost

[6] https://www.datacamp.com/courses/extreme-gradient-boosting-with-xgboost

[7] http://xgboost.readthedocs.io/en/latest/model.html

[8] http://xgboost.readthedocs.io/en/latest/python/python_intro.html

[9] https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html