

1. Kratak opis problema

U ovom projektu sam radio na problemu predikcije brzine usvajanja ljubimaca. Za to, koristio sam dataset koji nam je prilozen u projektu.

Cilj projekta je da napravim model koji treba da predvidi koliko brzo ce neki ljubimac biti usvojen, da li istog dana, u roku od 7 dana, u roku od 30, u roku od 90 dana ili ni posle 100.

Ukratko, zadatak je vrlo slozen i kompleksan, jer je jako tesko predvideti kada ce tacno, u kom vremenskom periodu, neki ljubimac biti usvojen.

U cilju pravljenja sto boljeg modela, radjene su razne analize na podacima, tj kako neki od podataka uticu na odluku kada ce neki ljubimac biti usvojen, a posle sam pravio modele koji koriste te podatke, kako bi probali da pronadju neke obrasce, koji bi pomogli da se odredi brzina usvajanja.

2. Informacije o podacima

Koristio sam prilozeni dataset, koji u sustini sadrzi oko 15000 podataka o ljubimcima u Maleziji. Sastoji se iz 3 skupa podataka, dati u .csv ili .pdf formatu:

1. Prvi deo podataka je tabularan, i predstavlja razne karakteristike ljubimaca, date kao numericke ili kategoricke vrednosti
2. Drugi deo podataka je isto tabularan, ali odvojen u drugi .csv, preprocesiran za koriscenje
3. Treci deo su slike ljubimaca, koje mozemo povezati preko PetID

Glavna target kolona je AdoptionSpeed, koja pokazuje koliko je brzo neki ljubimac usvojen:

Ova vrednost ima pet kategorija (od 0 do 4):

- 0 – ljubimac je usvojen istog dana,
- 1 – usvojen u roku od 7 dana,
- 2 – usvojen u roku od 30 dana,
- 3 – usvojen u roku od 90 dana,
- 4 – nije usvojen ni posle 100 dana.

Prilozeni dataset (prvi deo podataka) sam dalje spojio sa dodatnim .csv fajlom koji sadrzi rezultate sentiment analize ljubimaca, a to spajanje sam uradio pomocu PetID, pa sam dobio 2 nove kolone, SentimentScore (raspolozanje od -1 do 1) i SentimentMagnitude (jacina emocije).

Zatim sam proverio da li u podacima ima nedostajucih vrednosti. Manji broj praznih polja su imali Name, Description, SentimentScore i SentimentMagnitude, te sam obrisao redove u kojima imamo ova prazna polja.

Zatim sam u datasetu izdvojio samo ljubimce koji imaju ≥ 3 slike, jer zelim da imam bar 3 ulaza za CNN deo modela kasnije.

Na kraju sam dobio oko ~8700 ljubimaca koji zadovoljavaju preduslove, ali sam uzeo samo 2500 podataka da bi napravio dataset, jer je AdoptionSpeed = 0 retka, i ne zelim da imam „unbalanced“ podatke.

Najvise se pokazalo da je najvise ljubimaca u kategorijama 2,3,4, tako da uglavnom ljubimci uglavnom cekaju oko mesec dana do 3 meseca na usvajanje.

Od toga sam napravio train skup velicine 2000, validation skup velicine 250, i test skup velicine 250.

U kolonama se nalaze informacije poput:

- Type – tipivotinje (pas = 1, macka = 2)
- Age – starost ljubimca (data u mesecima)

- Breed1, Breed2 – glavna i sporedna rasa
- Gender – pol
- Color1, color2, color3 – boje
- MaturitySize – velicina
- FurLength – duzina krvana
- Vaccinated, Dewormed, Sterilized, Health – vakcinacija i generalne zdravstvene osobine ljubimca
- Quantity – broj ljubimaca
- State – drzava u Maleziji
- RescuerID – ko je spasio ljubimca
- Fee – Naknada za usvajanje
- PhotoAmt i VideoAmt – broj slika i video snimaka
- Description – tekstualni opis
- Img1, img2, img3 – putanje do slika svakog ljubimca

3. Analiza podataka

Uradjeno je 10 analiza podataka.

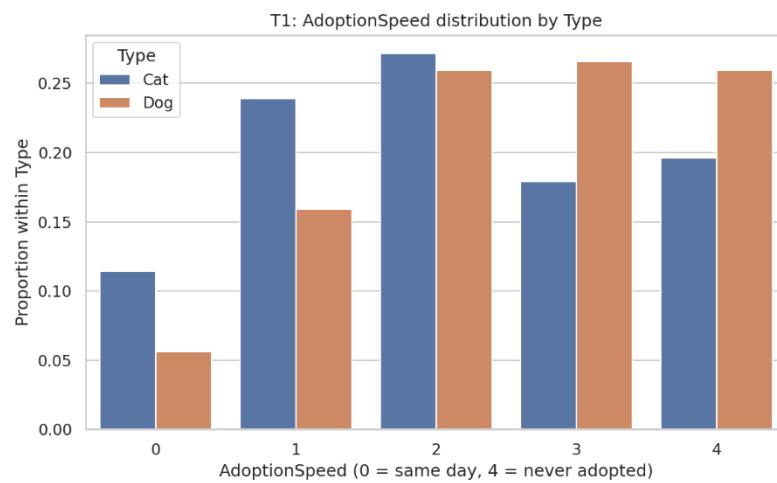
Analiza 1: AdoptionSpeed po tipu zivotinje

Na ovom grafikonu sam prikazao kako se brzina usvajanja razlikuje izmedju pasa i macaka.

Na x osi su ciljne kategorije (brzine usvajanja), a y osa predstavlja proporciju za tip ljubimca po kategoriji, gde jedan tip ljubimca (npr pas) u sumi svih kategorija brzine usvajanja ima vrednost 1.

Mozemo primetiti da je za macke veci proporcionalni ideo u klasama 0 i 1 – u odnosu na pse. To znaci da u ovom konkretnom slucaju, uzevsi samo tip ljubimca u oznici, veca je verovatnoca da ce macka biti usvojena isti dan ili u roku od 7 dana u odnosu na psa (u tom istom vremenskom periodu).

Ali mozemo primetiti da ideo usvajanja macaka opada u klasama 3,4, i da je tu pas dominantniji, znaci da ako je pas u pitanju, veca je verovatnoca da ce pas biti usvojen malo sporije nego kada je u pitanju macka.



Analiza 2: Starost u odnosu na AdoptionSpeed

U ovoj analizi sam htio da vidim da li starost ljubimca utice na brzinu usvajanja.

Prvo sam konvertovao starost iz meseci u godine ($AgeYears = Age / 12$), a zatim sam odsekao ekstremne vrednosti na 99.9 percentilu ($AgeYearsClipped$), da bi dijagram bio citljiv i da ga par vrlo starih ljubimaca ne bi razvuklo po y-osi.

Na grafiku se vidi da je vecina ljubimaca u svim klasama usvajanja mladja od 1 godine.

Gomila tacaka je nagomilana izmedju 0 i 1 godine za sve vrednosti AdoptionSpeed.

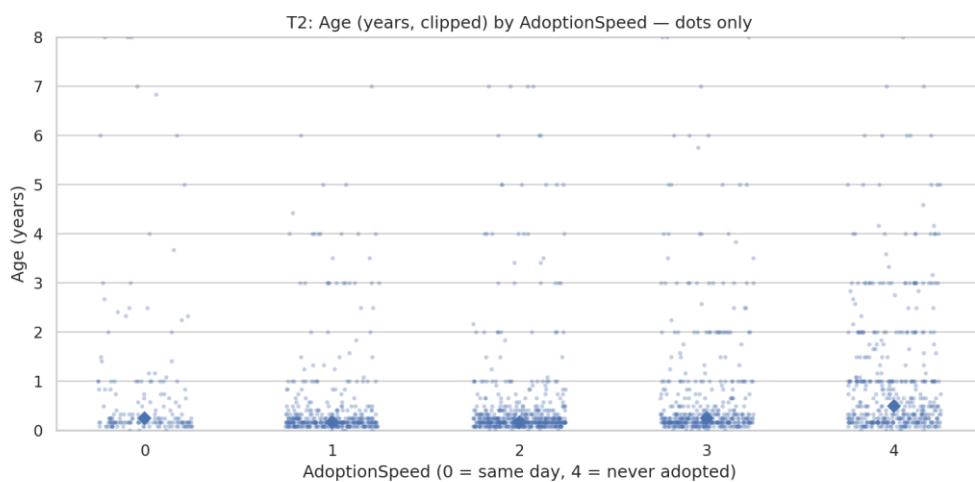
Diamanti na grafiku prikazuju median po svakoj klasi AdoptionSpeed, i tu mozemo videti da:

- za klase 1 i 2 median starosti je vrlo nizak (oko 2-3 meseca),
- za klasu 0 i 3 median je malo visi (oko 3-4 meseca)
- za klasu 4 (nikad usvojen) median je primetno veci (oko 6 meseci).

Iz ovoga mozemo zakljuditi da postoji neki blag trend, stariji ljubimci se malo sporije usvajaju od ljubimaca koji su starosti 2-3 meseca.

Medjutim, mozemo primetiti da je veliki broj ljubimaca koji su stariji od 5 godina takodje usvojeni prvi dan kada su dosli, ili 7 dana, 30 dana, ili 3 meseca nakon sto su dosli, kao i da nisu nikad usvojeni.

To znači da starost nije jedini faktor, ali realno može se reci da vrlo mlađi ljubimci imaju malo bolje sanse da budu brze usvojeni, a stariji će možda cekati malo duže.



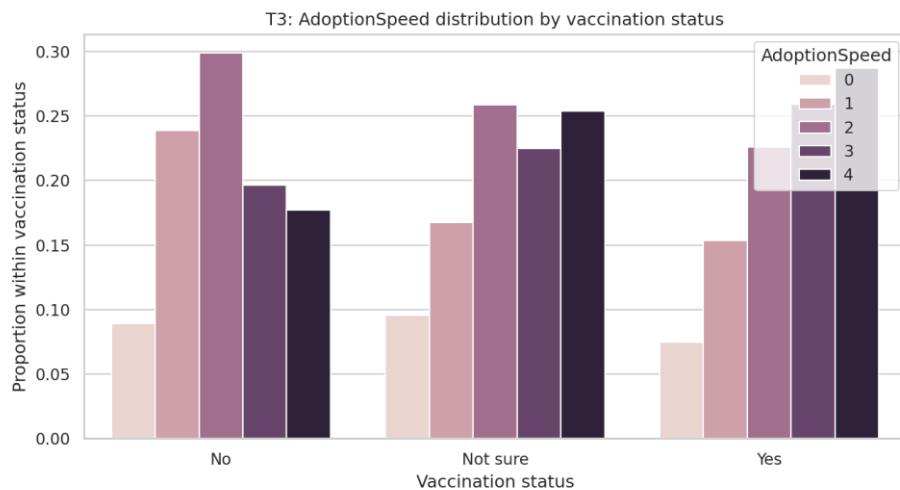
Analiza 3: Vakcinacija i brzina usvajanja

U ovoj analizi sam gledao kako se raspored AdoptionSpeed vrednosti menja u zavisnosti od statusa vakcinacije (Yes / No / Not sure).

Očekivao sam da će vakcinisani ljubimci biti brže usvojeni nego oni nevakcinisani, ali slika zapravo pokazuje suprotno:

- Kod vakcinisanih ljubimaca udeo kategorije 4 (nikad usvojen) je najvisi.
- Kod nevakcinisanih je više ljubimaca u srednjim brzim kategorijama (1 i 2), a udeo onih koji nikad nisu usvojeni je nizi nego kod vakcinisanih.
- Grupa Not sure je negde izmedju, bez jasnog efekta

Zaključak je da u ovom datasetu status vakcinacije ne pomaze da ljubimac bude brže usvojen, pa cak deluje da je efekat blago negativan. Verovatno ljudi vise gledaju druge osobine (starost, izgled, tip životinje), a vakcinaciju racunaju kao nesto sto mogu da srede kasnije ako se odluce da usvoje ovog ljubimca.



Analiza 4: Velicina i duzina krvna u odnosu na brzinu usvajanja

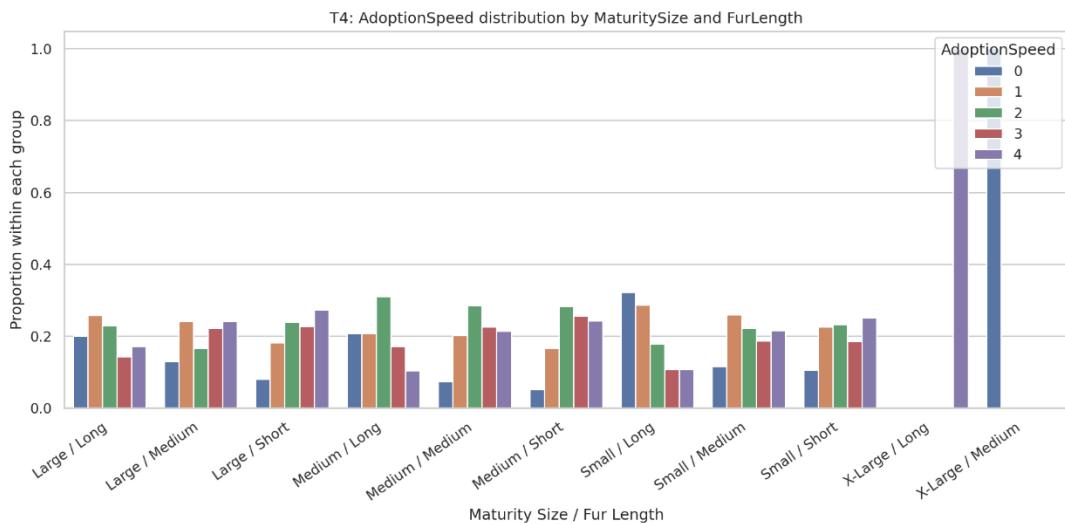
U ovoj analizi sam htio da vidim da li kombinacija velicine ljubimca i duzine krvna utice na brzinu usvajanja.

Zbog toga sam za svaku zivotinju napravio novu kategoriju Size_Fur (npr. Small / Long, Medium / Short, Large / Medium), i onda sam gledao raspodelu AdoptionSpeed vrednosti unutar svake te grupe.

Sa grafika se moze primetiti da se mali ljubimci sa dugom dlakom samo cesto usvajaju vrlo brzo (AdoptionSpeed = 0), a najmanje u kategoriji 4, sto znaci da se ova kombinacija generalno brzo usvaja. Takodje, malo se izdvajaju kombinacije Medium / Medium (pas srednje velicine sa srednjom duzinom krvna) i Medium / short, u kome se moze primetiti da ljubimci srednje velicine sa krakom dlakom se retko usvajaju isti dan, i procentualno vise pripadaju klasi 3 i 4.

Slicno mozemo videti i u kombinaciji velikog psa i kratke dlake.

Kod ostalih kombinacija, raspodele su prilicno slicne, nema nekih specijalnih kombinacija koje se izdvajaju i prave razliku.



Analiza 5: SentimentScore i SentimentMagnitude

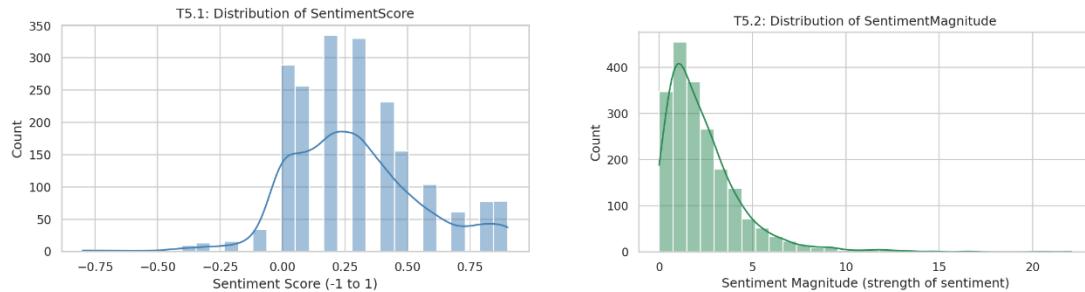
Prvo sam analizirao kako izgleda raspodela SentimentScore vrednosti, i video da je vecina opis pozitivna, uglavnom se skor kreće oko 0.25.

To znači da vecina ljudi lep opis za ljubimca.

Zatim sam analizirao SentimentMagnitude vrednosti, za jacinu emocije u tekstu.

Raspodela je skewed, najviše opisa ima malu jacinu, a blagi blok oglasa ima jako velike vrednosti.

To znači da je vecina opisa relativno malo emotivna, a redje poneki oglas je duzi i pun emocija.



Zatim sam posmatrao da li ljubimci koji se brže usvoje imaju pozitivnije opise.

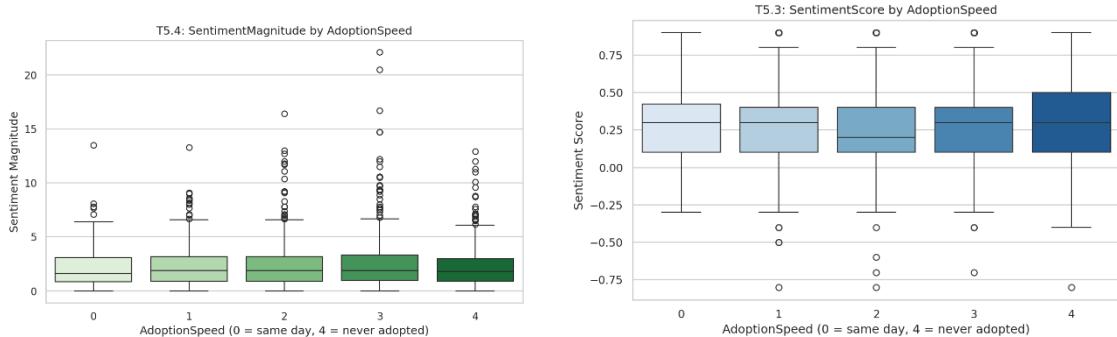
Slike pokazuju da su median SentimentScore vrednosti slicne za sve klase AdoptionSpeed – svi su blago pozitivni.

Nema jasnog trenda da su brže usvojeni ljubimci znacajno vise „pozitivno opisani“ u odnosu na one koji dugo cekaju.

Zaključak: pozitivnost teksta sama po sebi ne objasnjava mnogo brzinu usvajanja.

Osim toga, gledao sam da li duzi/emotivniji opisi pomazu da se ljubimac brže usvoji.

Vidi se blagi trend da ljubimci sa sporijim usvajanjem (klase 3 i 4) imaju **malo veće median vrednosti magnitude**, ali su raspodele jako preklapljene, ali ni to nije dovoljno jak signal da jasno razdvoji klase.



Analiza 6: SentimentScore i SentimentMagnitude

U ovoj analizi sam htio da kvantitativno proverim ono sto sam vec video na grafikima iz analize 5 – da li sentiment opisa stvarno ima ikakve veze sa AdoptionSpeed.

Zato sam izracunao eta-squared (η^2) efekat:

- $\eta^2(\text{SentimentScore} \sim \text{AdoptionSpeed}) = 0.0022$
- $\eta^2(\text{SentimentMagnitude} \sim \text{AdoptionSpeed}) = 0.0044$

Vrednosti blizu 0 nemaju gotovo nikad efekat, a vec od 0.01 se smatra malim efektom.

Posto su moje vrednosti manje od 0.01, imaju vrlo mali efekat na brzinu usvajanja, pogotovo SentimentScore, dok SentimentMagnitude mozda ima vrlo mali uticaj.

Analiza 7: Statisticki momenti za par feature-a

Za Age, Fee, sentimentScore i SentimentMagnitude sam izracunao srednju vrednost, varijansu, skewness (asimetriju) i kurtosis (tezinu repova).

Feature: AgeYears

Mean : 0.7965, vecina ljubimaca je mladje od godinu dana

Variance : 2.2020, ima rasipanja, jer postoji i manji broj starijih ljubimaca

Skewness : 4.2667, veliki, raspodela je nagnuta u desno (right skew), vecina podataka je levo

Kurtosis : 26.6197, par vrlo starih ljubimaca pravi rep, teski repovi

Feature: Fee

Mean : 20.8159, prosečno oko 20, ali vecina ima 0

Variance : 6075.4065, ogromna varijansa, neki oglasi imaju Fee mnogo veci od proseka

Skewness : 5.7920, jak right skewness, vecina oglasa ima mali ili Fee=0, a mali broj ima jako veliki Fee

Kurtosis : 42.7674, ekstremno teski repovi, par oglasa sa ogromnim Fee

Feature: SentimentScore

Mean : 0.2901, opisi su blago pozitivni, kao i prethodno zaključeno

Variance : 0.0704, mala varijansa, nema velike rasutosti

Skewness : 0.2905, mali skewness, nesto vise pozitivnih nego negativnih skorova

Kurtosis : 0.4689, nema jakih repova

Feature: SentimentMagnitude

Mean : 2.3743, prosečno

Variance : 4.7010, postoji dobra varijansa u duzini i emotivnosti opisa

Skewness : 2.5342, right skew, mali, vecina oglasa ima malu do srednju magnitudu

Kurtosis : 11.4397 – srednje teski repovi, par oglasa sa bas velikom magnitudom.

Analiza 8: Statisticki momenti za AdoptionSpeed klasu

Za Age, Fee, sentimentScore i SentimentMagnitude sam izracunao srednju vrednost, varijansu, skewness (asimetriju) i kurtosis (tezinu repova).

Feature: AgeYears

AdoptionSpeed = 0: mean=0.8388, var=2.3843, skew=3.2443, kurt=10.7273

AdoptionSpeed = 1: mean=0.5956, var=1.3748, skew=4.1389, kurt=21.5625

AdoptionSpeed = 2: mean=0.5975, var=1.5656, skew=4.2162, kurt=19.8458

AdoptionSpeed = 3: mean=0.8244, var=2.9932, skew=5.5823, kurt=43.1312

AdoptionSpeed = 4: mean=1.1568, var=2.6147, skew=2.7830, kurt=9.3787

Feature: SentimentMagnitude

AdoptionSpeed = 0: mean=2.1530, var=3.5465, skew=2.1369, kurt=7.9702

AdoptionSpeed = 1: mean=2.3701, var=3.5942, skew=1.5372, kurt=3.6979

AdoptionSpeed = 2: mean=2.4068, var=4.6831, skew=2.1672, kurt=7.2678

AdoptionSpeed = 3: mean=2.5886, var=7.0949, skew=3.0833, kurt=14.4075

AdoptionSpeed = 4: mean=2.2116, var=3.6988, skew=2.0069, kurt=5.9617

AgeYears

Video sam da je prosečna starost najniza za klase 1 i 2 (oko 0.6 godina), veca za klasu 3 (0.82), a najveca za klasu 4 (oko 1.16 godina).

To znači da ljubimci koji se najsporije usvajaju u proseku jesu malo stariji.

Skewness i kurtosis su veliki u svim klasama, što znači da u svakoj klasi ima gomila jako mladih i mali broj mnogo starijih ljubimaca.

SentimentMagnitude

Prosecne vrednosti se krecu izmedju 2.15 i 2.59, razlike su male.

Klasa 3 ima malo vecu varijansu i jacu desnu asimetriju, ali svi imaju desni rep i teske repove.

Zakljucak: nema jasnog i jakog obrasca po klasama, sto se lepo uklapa sa prethodnim rezultatima da sentiment nije bas dobar prediktor za brzinu usvajanja ljubimca.

Analiza 9: Osnovne statistike slika

U ovoj analizi sam htio da proverim osnovne statistike slika, da bih znao da li su slike sличnih dimenzija i osvetljenja, i da li je normalno da ih sve resize na isti format.

Uzeo sam prvih 500 slika iz kolone img1 (ovo sam ranije generisao) i za svaku sam izracunao visinu, sirinu, odnos sirina/visina i prosecno osvetljenje.

Dobio sam:

Visina: prosecak oko 397 px, od 84 do 640

Sirina: prosecak oko 397.5 px, od 68 do 640

Aspect ratio (w/h): prosecak 1.044, std ≈ 0.355

Zakljucak: vecina slika je priblizno kvadratna, ali ima i malo nepravilnijih formata.

Mean intensity: prosecak oko 119.5 (na skali 0–255), sa std ≈ 27.3

Slike su u proseku umereno osvetljene, ima i tamnijih i svetlijih, ali nema ekstremno mračnih ili svetlih.

Zakljucak:

slike su uglavnom srednje rezolucije, prilicno sличnih dimenzija i osvetljenja, tako da ima smisla da ih sve rizerujem na 128x128 i da ne radim neke dodatne korekcije osvetljenja pre treniranja modela.

Analiza 10: Prosecna slika psa i prosecna slika macke

I za macku i za psa sam uzeo po prvih 300 slika iz img1 path koji sam prethodno kreirao, smanjio ih na istu velicinu 128x128, pretvorio u RGB i onda izracunao piksel po piksel prosecak. To pravi mutnu sliku, ali ta slika predstavlja prosecni izgled klase. Zanimljivo je da se dobija poprilično dobra slika za psa kada povecamo na prvih 1000 slika.

Rezultat:

- Prosecna slika psa je sivkasta sa blago tamnim centrom, vidi se da su psi najcesce u sredini slike.
- Profesna slika macke je vrlo slicna, ali moza malo ravnomernija, ali i dalje ne potpuno jasna. Moza mogu da se prepoznaju usi macke, koje su izduzenje nego kod psa, to bi bilo nego moje zapazanje.

Zakljucak: pojedicne slike macaka i pasa mogu biti raznih formata, ali prosecan stil kao sto je pozadina, boja, kadar im je generalno slican. Model nece moci da pravi veliku razliku po boji, vec mora da uci detaljnije vizuelne obrasce na slikama.

Average Cat image (n=300)



Average Dog image (n=300)



4. Zakljucak iz analize podataka

Problem koliko brzo se ljubimci usvajaju i koji faktori najvise uticu na brzinu usvajanja je izuzetno komplikovan. Napravio sam 10 analiza da bi probao da pronadjem neki faktor, i problem je izuzetno tezak. Generalno, moze se primetiti da najvise ljubimaca zavrsi u srednjim ili vecim klasama, tako da je brzo usvajanje dosta redje.

Samim tim sto je brzo usvajanje dosta redje, nemamo puno podataka sa kojima mozemo da radimo da bi uspesno detektovali ovu klasu, sto ce verovatno uticati kasnije na tacnost modela.

Sa strane tabularnih podataka, pokazalo se da tip zivotinje i starost imaju uticaja, macke se brze usvajaju nego psi, i mladji ljubimci oko 2-3 meseca imaju nesto bolje sanse da nadju dom nego oni stariji.

Medjutim, iako nisam ocekivao ovakve rezultate, ima dosta starijih ljubimaca koji se usvoje brzo, pa ovi efekti nisu presudni, vec samo blagi trendovi. Velicina i drugina krvna mogu takodje da imaju neku ulogu, kao sto smo videli da mali ljubimci sa duzom dlakom imaju vece sanse da budu brze udomljeni, ali vecina kombinacija ne pravi neku jasnu razliku.

Analiza oko vakcinacije, fee-a i sentimenta pokazuju da ovi faktori imaju vrlo slab efekat i prilicno nemaju jasan obrazac. Cak kontaintuitivno, vakcinisani ljubimci se ne usvajaju brze, sto mozda moze da se poveze da stariji ljubimci su cesce vec vakcinisani, a mladji ljubimci nisu, a vec smo videli da mladji ljubimci imaju vece sanse da budu brze usvojeni, to bi bilo neko moje zapazanje.

Sa druge strane, cak ni sentiment opisa ne igra veliku ulogu, dok bi neko logicno objasnjenje bilo da lep opis = pozitivno misljenje = brze usvajanje, ali ocigledno nije tako. Takodje i fee nema nekog velikog uticaja, i cak je nelogично да ako se placa veliki fee – zar to ne bi znacilo da ljubimac imati manje sanse da bude brze usvojen? Ali mozda i ovo moze da se poveze sa Age, jer za mладje ljubimce mozda mora da se plati neki fee, koji su vlasnici cesto spremni da daju ako im se pas dopadne.

Na osnovu ovih podataka, nemamo neki bas doabz zakljucak da neki feature mnogo utice na to da li ce ljubimac biti brze usvojen, jer veliku igru u tome imaju subjektivni osecaji oko ljubimca, cak ljubimci koji imaju „losije karakteristike“ i po logici bi trebali da budu sporije usvojeni, nije tako jer je moguce da veliku ulogu imaju same emocije vlasnika koji bi usvojio psa, i ukoliko vidi nekog ljubimca u losijem stanju, izazvace takav efekat u coveku da ce mu biti zao i mozda usvojiti bas njega, sto po statistici i pravilima nema mnogo smisla.

Sa strane slika, osnovne statistike pokazuju da su fotografije slicnih dimenzija i osvetljenja, pa ima smisla da se sve resize-uju na isti format i koriste u nekom konvolucionom modelu. Prosečne slike psa i macke ispadaju vrlo slicne sivkaste mrlje, sto znaci da model nece moci da razlikuje tip zivotinje po nekoj trivijalnoj razlici u pozadini ili boji, vec ce morati da nauci realne vizuelne obrascce (oblik tela, usi, nuska itd.).

Ukratko, iz ove analize bih izdvojio da ce za predikciju brzine usvajanja najvise znaciti kombinacija osnovnih karakteristika ljubimca (tip, starost, velicina, krvna, mozda jos broj slika, rasa i slicno) zajedno sa informacijom sa slika, dok tekstualni opis i sentiment deluju kao sporedni faktori, bez nekog velikog uticaja. Ovi uvidi su mi bili polaziste za dizajn dalje arhitekture modela – prvo cist tabularni model, a zatim multimodalni model koji kombinuje tabularne podatke i slike.

5. Opis resenja

Nakon analize podataka, odlucio sam da pristupim problemu na 2 nacina:

1. Tabularni model: prvo sam napravio tabularni model koji koristi sve numericke i kategorickie osobine (i setiment). Ovo sluzi kao osnovna, pocetna referencia za to koliko se moze izvuci iz tabela, bez slika.
2. Multimodalni modeli: zatim sam napravio vise verzija modela koji kombinuju tabularne podatke, i slike ljubimaca preko CNN grane, i imao sam 3 slike po ljubimcu.

Cilj mi je bio da vidim da li dodavanje slika poboljsava tacnost u odnosu na plain tabularni model, i kako uticu 9 kanala, 3 odvojene slike, RGB vs grayscale, augmentacija na performanse.

Za sve modele je koriscen isti train/validation/test split (2000 / 250 / 250), adam optimized sa LR 0.001, Early stopping na val loss, i reduceLR on plateau, da bi sprecio overfitting.

6. Opis Arhitekture modela i 7. opis eksperimenata

Model 1: - Tabularni

Ovo je model koji koristi samo tabularne podatke. Odavde sam krenuo, i gledao koliku tacnost mogu da dobijem bez slika.

Sto se tice preprocesiranja, radio sam log transformaciju za Age, Fee i SentimentMagnitude, zbog right skewness. Dalje, koristio sam numericke i kategorickie podatke, a izbacio sam neke feature kao sto su PetId jer ne uticu na performanse modela.

Za numericki deo je koriscen StandardScaler, a za kategoricki deo OneHotEncoder.

Naravno, preprocess se fittuje samo na train skupu.

Target je AdoptionSpeed, od 0 do 4.

Arhitektura modela:

[INFO] Features processed. Input shape: (1999, 65)		
Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	8,448
batch_normalization (BatchNormalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2,080
batch_normalization_2 (BatchNormalization)	(None, 32)	128
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 5)	165

Rezultat:

Tacnost na test skupu je oko 33-34%. Model hvata neke osnovne obrasce, da stariji ljubimci se sporiju usvajaju i slicno, ali ne uspeva da predvidi tacnu vremensku kategoriju dovoljno dobro. Pogotovo mu je tesko da prediktuje klasu 0, verovatno zbog unbalanced dataseta.

```
Epoch 20/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3660 - loss: 1.4292 - val_accuracy: 0.3506 - val_loss: 1.4599 - learning_rate: 0.0010
Epoch 21/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3784 - loss: 1.4017 - val_accuracy: 0.3426 - val_loss: 1.4619 - learning_rate: 0.0010
Epoch 22/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3848 - loss: 1.4320 - val_accuracy: 0.3386 - val_loss: 1.4662 - learning_rate: 0.0010
Epoch 23/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4007 - loss: 1.4156 - val_accuracy: 0.3466 - val_loss: 1.4663 - learning_rate: 0.0010
Epoch 24/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3723 - loss: 1.4201 - val_accuracy: 0.3386 - val_loss: 1.4658 - learning_rate: 5.0000e-04
Epoch 25/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3961 - loss: 1.4074 - val_accuracy: 0.3386 - val_loss: 1.4633 - learning_rate: 5.0000e-04
Epoch 26/100
63/63 ━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3888 - loss: 1.3930 - val_accuracy: 0.3426 - val_loss: 1.4598 - learning_rate: 5.0000e-04
Epoch 27/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4213 - loss: 1.3651 - val_accuracy: 0.3386 - val_loss: 1.4620 - learning_rate: 5.0000e-04
Epoch 28/100
63/63 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3809 - loss: 1.4073 - val_accuracy: 0.3267 - val_loss: 1.4660 - learning_rate: 5.0000e-04

=====
FINAL TEST EVALUATION
=====
8/8 ━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3525 - loss: 1.4161
Test Accuracy: 34.40%

Process finished with exit code 0
```

Model 2 – Multimodalni model, 3 slike spojene u 9 kanala

Ovo je 1. multimodalna verzija. Hteo sam da dodam slike, ali sto jednostavnije, zato sam 3 RGB slike spojio u jedan tensor oblika (128, 128, 9) i na njega primenio CNN.

Preprocesiranje podataka je uradjeno isto kao i u modelu 1, a image putanje se koriste za slike.

Za image deo sam imao:

Ulez: image_input oblika (128, 128, 9)

Rescaling

Conv2D (32, 3x3, ReLU, padding = 'same') i MaxPool

Conv2D (64, 3x3, ReLU, padding = 'same') i MaxPool

Conv2D (128, 3x3, ReLU, padding = 'same') i MaxPool

GlobalAveragePooling2D, Dense(64, ReLU).

Za tabular deo:

Dense (128, ReLU), batch normalization, dropout 0.3

Dense (64, ReLU)

Merge:

Concatenate (tabelar_features, image_features)

Dense (64, Relu) + Dropout (0.3)

Izlaz: Dense (5, softmax)

Dodatno, koristio sam class_weight = balanced, da bi ublazio nebalansiran target. Mada ovo nije pokazalo neke mnogo bolje rezultate, ali u praksi bi ovo trebalo da pomogne u slicnim situacijama.

[INFO] Tabular Input Shape: (1999, 65)																																																																																
[INFO] Class weights: {0: 2.3797619047619047, 1: 1.0147208121827411, 2: 0.7543396226415094, 3: 0.890423162583519, 4: 0.8729257641921397}																																																																																
[INFO] Creating TF Datasets...																																																																																
Model: "functional"																																																																																
<table border="1"><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th><th>Connected to</th></tr></thead><tbody><tr><td>image_input (InputLayer)</td><td>(None, 128, 128, 9)</td><td>0</td><td>-</td></tr><tr><td>rescaling (Rescaling)</td><td>(None, 128, 128, 9)</td><td>0</td><td>image_input[0][0]</td></tr><tr><td>conv2d (Conv2D)</td><td>(None, 128, 128, 32)</td><td>2,624</td><td>rescaling[0][0]</td></tr><tr><td>max_pooling2d (MaxPooling2D)</td><td>(None, 64, 64, 32)</td><td>0</td><td>conv2d[0][0]</td></tr><tr><td>conv2d_1 (Conv2D)</td><td>(None, 64, 64, 64)</td><td>18,496</td><td>max_pooling2d[0][0]</td></tr><tr><td>tabular_input (InputLayer)</td><td>(None, 65)</td><td>0</td><td>-</td></tr><tr><td>max_pooling2d_1 (MaxPooling2D)</td><td>(None, 32, 32, 64)</td><td>0</td><td>conv2d_1[0][0]</td></tr><tr><td>dense_1 (Dense)</td><td>(None, 128)</td><td>8,448</td><td>tabular_input[0][0]</td></tr><tr><td>conv2d_2 (Conv2D)</td><td>(None, 32, 32, 128)</td><td>73,856</td><td>max_pooling2d_1[0][0]</td></tr><tr><td>batch_normalization (BatchNormalizatio...</td><td>(None, 128)</td><td>512</td><td>dense_1[0][0]</td></tr><tr><td>max_pooling2d_2 (MaxPooling2D)</td><td>(None, 16, 16, 128)</td><td>0</td><td>conv2d_2[0][0]</td></tr><tr><td>dropout (Dropout)</td><td>(None, 128)</td><td>0</td><td>batch_normalizatio...</td></tr><tr><td>global_average_poo... (GlobalAveragePool...)</td><td>(None, 128)</td><td>0</td><td>max_pooling2d_2[0][0]</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 64)</td><td>8,256</td><td>dropout[0][0]</td></tr><tr><td>dense (Dense)</td><td>(None, 64)</td><td>8,256</td><td>global_average_poo...</td></tr><tr><td>concatenate (Concatenate)</td><td>(None, 128)</td><td>0</td><td>dense_2[0][0], dense[0][0]</td></tr><tr><td>dense_3 (Dense)</td><td>(None, 64)</td><td>8,256</td><td>concatenate[0][0]</td></tr><tr><td>dropout_1 (Dropout)</td><td>(None, 64)</td><td>0</td><td>dense_3[0][0]</td></tr><tr><td>dense_4 (Dense)</td><td>(None, 5)</td><td>325</td><td>dropout_1[0][0]</td></tr></tbody></table>	Layer (type)	Output Shape	Param #	Connected to	image_input (InputLayer)	(None, 128, 128, 9)	0	-	rescaling (Rescaling)	(None, 128, 128, 9)	0	image_input[0][0]	conv2d (Conv2D)	(None, 128, 128, 32)	2,624	rescaling[0][0]	max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0	conv2d[0][0]	conv2d_1 (Conv2D)	(None, 64, 64, 64)	18,496	max_pooling2d[0][0]	tabular_input (InputLayer)	(None, 65)	0	-	max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_1[0][0]	dense_1 (Dense)	(None, 128)	8,448	tabular_input[0][0]	conv2d_2 (Conv2D)	(None, 32, 32, 128)	73,856	max_pooling2d_1[0][0]	batch_normalization (BatchNormalizatio...	(None, 128)	512	dense_1[0][0]	max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 128)	0	conv2d_2[0][0]	dropout (Dropout)	(None, 128)	0	batch_normalizatio...	global_average_poo... (GlobalAveragePool...)	(None, 128)	0	max_pooling2d_2[0][0]	dense_2 (Dense)	(None, 64)	8,256	dropout[0][0]	dense (Dense)	(None, 64)	8,256	global_average_poo...	concatenate (Concatenate)	(None, 128)	0	dense_2[0][0], dense[0][0]	dense_3 (Dense)	(None, 64)	8,256	concatenate[0][0]	dropout_1 (Dropout)	(None, 64)	0	dense_3[0][0]	dense_4 (Dense)	(None, 5)	325	dropout_1[0][0]
Layer (type)	Output Shape	Param #	Connected to																																																																													
image_input (InputLayer)	(None, 128, 128, 9)	0	-																																																																													
rescaling (Rescaling)	(None, 128, 128, 9)	0	image_input[0][0]																																																																													
conv2d (Conv2D)	(None, 128, 128, 32)	2,624	rescaling[0][0]																																																																													
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0	conv2d[0][0]																																																																													
conv2d_1 (Conv2D)	(None, 64, 64, 64)	18,496	max_pooling2d[0][0]																																																																													
tabular_input (InputLayer)	(None, 65)	0	-																																																																													
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_1[0][0]																																																																													
dense_1 (Dense)	(None, 128)	8,448	tabular_input[0][0]																																																																													
conv2d_2 (Conv2D)	(None, 32, 32, 128)	73,856	max_pooling2d_1[0][0]																																																																													
batch_normalization (BatchNormalizatio...	(None, 128)	512	dense_1[0][0]																																																																													
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 128)	0	conv2d_2[0][0]																																																																													
dropout (Dropout)	(None, 128)	0	batch_normalizatio...																																																																													
global_average_poo... (GlobalAveragePool...)	(None, 128)	0	max_pooling2d_2[0][0]																																																																													
dense_2 (Dense)	(None, 64)	8,256	dropout[0][0]																																																																													
dense (Dense)	(None, 64)	8,256	global_average_poo...																																																																													
concatenate (Concatenate)	(None, 128)	0	dense_2[0][0], dense[0][0]																																																																													
dense_3 (Dense)	(None, 64)	8,256	concatenate[0][0]																																																																													
dropout_1 (Dropout)	(None, 64)	0	dense_3[0][0]																																																																													
dense_4 (Dense)	(None, 5)	325	dropout_1[0][0]																																																																													
Epoch 10/30 63/63 4s 58ms/step - accuracy: 0.4000 - loss: 1.3774 - val_accuracy: 0.3460 - val_loss: 1.4831 - learning_rate: 0.0010																																																																																
Epoch 11/30 63/63 4s 62ms/step - accuracy: 0.3895 - loss: 1.3567 - val_accuracy: 0.3307 - val_loss: 1.5227 - learning_rate: 0.0010																																																																																
Epoch 12/30 63/63 4s 63ms/step - accuracy: 0.3872 - loss: 1.4096 - val_accuracy: 0.3187 - val_loss: 1.5416 - learning_rate: 0.0010																																																																																
Epoch 13/30 63/63 5s 71ms/step - accuracy: 0.4038 - loss: 1.3399 - val_accuracy: 0.3147 - val_loss: 1.5359 - learning_rate: 0.0010																																																																																
Epoch 14/30 63/63 5s 73ms/step - accuracy: 0.4273 - loss: 1.2862 - val_accuracy: 0.3307 - val_loss: 1.5543 - learning_rate: 0.0010																																																																																
Epoch 15/30 63/63 5s 82ms/step - accuracy: 0.4184 - loss: 1.2945 - val_accuracy: 0.3347 - val_loss: 1.5546 - learning_rate: 5.0000e-04																																																																																
Epoch 16/30 63/63 4s 58ms/step - accuracy: 0.4441 - loss: 1.2518 - val_accuracy: 0.3227 - val_loss: 1.5377 - learning_rate: 5.0000e-04																																																																																
Epoch 17/30 63/63 4s 63ms/step - accuracy: 0.4392 - loss: 1.2266 - val_accuracy: 0.3108 - val_loss: 1.5586 - learning_rate: 5.0000e-04																																																																																
Epoch 18/30 63/63 4s 68ms/step - accuracy: 0.4152 - loss: 1.2581 - val_accuracy: 0.3187 - val_loss: 1.5571 - learning_rate: 5.0000e-04																																																																																
=====																																																																																
FINAL TEST EVALUATION																																																																																
=====																																																																																
8/8 0s 33ms/step - accuracy: 0.3728 - loss: 1.4626																																																																																
Test Accuracy: 31.60%																																																																																
[INFO] Model saved.																																																																																
Process finished with exit code 0																																																																																

Rezultat:

Tacnost na testu je slična, ili malo bolja od modela 1. Cnn ogledno donosi dodatnu informaciju, ali spajanje 3 slike u 9 kanala je prilично grubo, i verovatno ne iskoriscava slike na najbolji nacin.

Model 3 – Multimodalni model sa time distributed CNN

Ovaj model je unapredjenje modela 2. Umesto da 3 slike lepim po kanalima, tretiram ih kao posebnu sekvencu duzine 3, i primenjujem isti CNN na svaku sliku posebno. Na kraju agregiram reprezentacije.

Glavna razlika u odnosu na model 2 je da je ulaz za slike obila (3, 128, 128, 3), i da svaka slika prolazi kroz isti CNN blok, sto bi trebalo da bude efikasnije.

Image deo:

Isto radimo scaling kao u modelu 2.

Zatim radimo:

TimeDistributed(Conv2D(32, 3, ReLU, padding='same')) + TimeDistributed(MaxPool)

TimeDistributed(Conv2D(64, 3, ReLU, padding='same')) + TimeDistributed(MaxPool)

TimeDistributed(Conv2D(128, 3, ReLU, padding='same')) + TimeDistributed(MaxPool)

TimeDistributed(GlobalAveragePooling2D) , dobijamo tensor (batch, 3, 128)

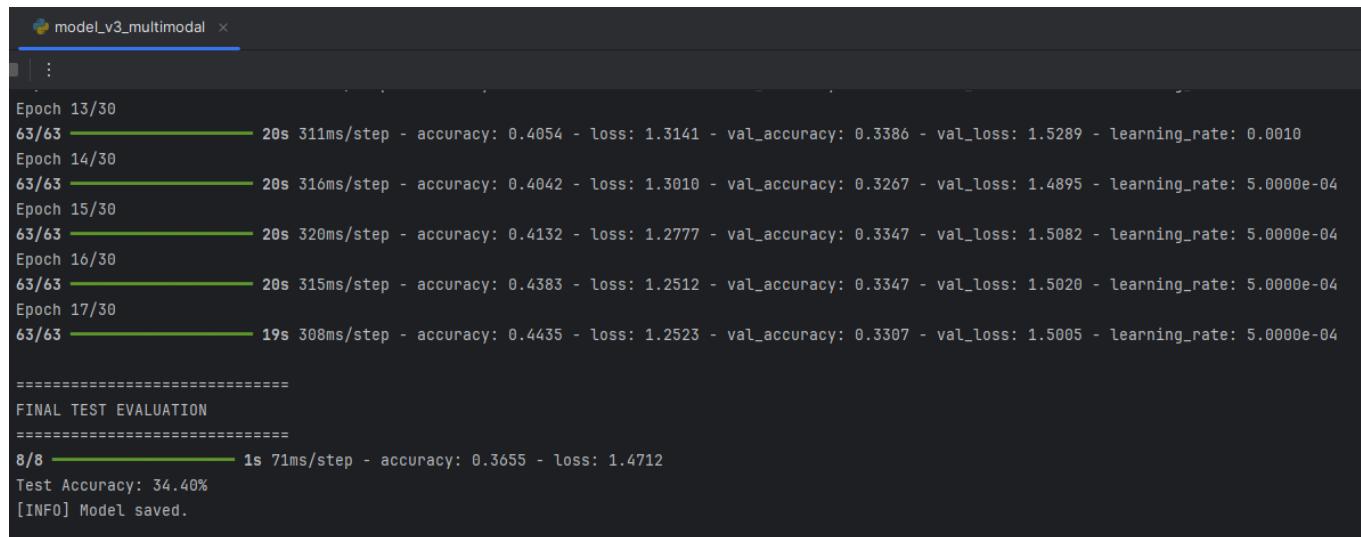
GlobalAveragePooling1D preko dimenzije slike

Tabular deo i merge deo su isti kao u modelu 2.

Rezultat:

Tacnost na test skpu je bila najcesce bolja od modela 2, i najcesce bila preko 0.35. Ovome doprinosi TimeDistributed pristup koji omogucava modelu da uci iste filtere za sve 3 slike, sto se pokazali kao najstabilniji pristup u ovim verzijama modela.

Accuracy je isao do 0.45. Validation accuracy uglavnom nije prelazio 0.36.



```
model_v3_multimodal ×
Epoch 13/30
63/63 20s 311ms/step - accuracy: 0.4054 - loss: 1.3141 - val_accuracy: 0.3386 - val_loss: 1.5289 - learning_rate: 0.0010
Epoch 14/30
63/63 20s 316ms/step - accuracy: 0.4042 - loss: 1.3010 - val_accuracy: 0.3267 - val_loss: 1.4895 - learning_rate: 5.0000e-04
Epoch 15/30
63/63 20s 320ms/step - accuracy: 0.4132 - loss: 1.2777 - val_accuracy: 0.3347 - val_loss: 1.5082 - learning_rate: 5.0000e-04
Epoch 16/30
63/63 20s 315ms/step - accuracy: 0.4383 - loss: 1.2512 - val_accuracy: 0.3347 - val_loss: 1.5020 - learning_rate: 5.0000e-04
Epoch 17/30
63/63 19s 308ms/step - accuracy: 0.4435 - loss: 1.2523 - val_accuracy: 0.3307 - val_loss: 1.5005 - learning_rate: 5.0000e-04
=====
FINAL TEST EVALUATION
=====
8/8 1s 71ms/step - accuracy: 0.3655 - loss: 1.4712
Test Accuracy: 34.40%
[INFO] Model saved.
```

Nadalje, eksperimentisao sam sa modelima 4 i 5 – sta se desava kada dodamo grayscale, i sta se desava kada dodamo augmentaciju.

Model 4 – Multimodalni TimeDistributed, Grayscale slike

Ovaj model je arhitektonski isti kao model 3, ali slike pretvaram u grayscale 1 kanal. Ideja mi je bila da posto vec imamo color parametre zadate u tabularnim podacima, RGB nije toliko bitan i stvara samo dodatno opterecenje za prediktor.

Promene u odnosu na model 3:

- Slike postaju (128, 128, 1)
- Naravno, vrsimo grayscaling.
- Ulaz je image_input oblika (3, 128, 128, 1)
- CNN arhitektura ostaje ista.

Rezultat:

Tacnost je slicna kao u Modelu 3, ili cak malo niza ponekad, oko 0.33 - 0.37%. Grayscale slike ocigledno nose dovoljno informacija da model ostane u istom rangu performansi, ali super stvar je sto dobijamo brze treniranje, uz vrlo slicno rezultate.

```
Run model_v4_multimodal ×
[Run] : 63/63 19s 300ms/step - accuracy: 0.4047 - loss: 1.3867 - val_accuracy: 0.3227 - val_loss: 1.4654 - learning_rate: 0.0010
↑ Epoch 10/30
↓ 63/63 19s 303ms/step - accuracy: 0.4063 - loss: 1.3747 - val_accuracy: 0.3187 - val_loss: 1.4448 - learning_rate: 0.0010
[Run] Epoch 11/30
[Run] 63/63 20s 317ms/step - accuracy: 0.4076 - loss: 1.3528 - val_accuracy: 0.3347 - val_loss: 1.4607 - learning_rate: 5.0000e-04
[Run] Epoch 12/30
[Run] 63/63 19s 299ms/step - accuracy: 0.4383 - loss: 1.3172 - val_accuracy: 0.3267 - val_loss: 1.4712 - learning_rate: 5.0000e-04
[Run] Epoch 13/30
[Run] 63/63 19s 298ms/step - accuracy: 0.4655 - loss: 1.2922 - val_accuracy: 0.3426 - val_loss: 1.4686 - learning_rate: 5.0000e-04
[Run] Epoch 14/30
[Run] 63/63 20s 322ms/step - accuracy: 0.4392 - loss: 1.2828 - val_accuracy: 0.3546 - val_loss: 1.4772 - learning_rate: 5.0000e-04
=====
FINAL TEST EVALUATION
=====
8/8 1s 62ms/step - accuracy: 0.3527 - loss: 1.4554
Test Accuracy: 33.60%
[INFO] Model saved.

Process finished with exit code 0
```

Model 5 – Multimodalni TimeDistributed, grayscale slike i dodatno augmentacija

U poslednjoj verziji sam dodao augmentaciju slika direktno u model. Svaka od 3 grayscale slike prolaze kroz random transformacije koje generalno cesto radimo, npr flip, rotacija, zoom, contrast, i to u minimalnim granicama jer ne zelimo da napravimo lose augmentirane podatke.

Cilj mi je bio da poboljsam generalizaciju, da ucinim CNN manje osetljivim na tacnu poziciju i osvetljenje ljubimaca, i da uvedemo jos dodatnih slika za treniranje, jer generalno nemamo puno podataka za krenuti.

Promene u odnosu na model 4:

Pre reskaliranja dodati su sledeci slojevi (TimeDistributed):

- RandomFlip("horizontal")
- RandomRotation(0.1)
- RandomZoom(0.1)
- RandomContrast(0.1)

Ostali delovi (tabular deo, CNN, merge) su identicni Modelu 4.

Rezultat:

Tacnost je bila vrlo slicna ostalim modelima, ali zanimljivo da je prosecno bila slicna modelu 3 i modelu 4.

Osigledno je da ovo nije pomoglo modelu, i prediktor nije postao bolji zbog ovoga.

Cinjenica je da augmentacija dodatno siri prostor primera, ovo otezava trening mreze koja se ionako trenira od nule. Mozda bi se augmentacija vise isplatila uz veci broj epoha, veci dataset, i malo boljim namestanjem.

Mada, po parametrima (accuracy, loss, val accuracy, val loss) u rangu je sa najboljim rezultatima.

```
model_v5_multimodal x
:
Epoch 11/30
63/63 18s 282ms/step - accuracy: 0.4169 - loss: 1.3457 - val_accuracy: 0.3546 - val_loss: 1.4716 - learning_rate: 0.0010
Epoch 12/30
63/63 18s 284ms/step - accuracy: 0.4533 - loss: 1.2840 - val_accuracy: 0.3267 - val_loss: 1.4700 - learning_rate: 5.0000e-04
Epoch 13/30
63/63 19s 302ms/step - accuracy: 0.4302 - loss: 1.2857 - val_accuracy: 0.3187 - val_loss: 1.4755 - learning_rate: 5.0000e-04
Epoch 14/30
63/63 18s 285ms/step - accuracy: 0.4499 - loss: 1.2771 - val_accuracy: 0.3386 - val_loss: 1.4831 - learning_rate: 5.0000e-04
Epoch 15/30
63/63 18s 285ms/step - accuracy: 0.4767 - loss: 1.2719 - val_accuracy: 0.3586 - val_loss: 1.4934 - learning_rate: 5.0000e-04
=====
FINAL TEST EVALUATION
=====
8/8 1s 61ms/step - accuracy: 0.3453 - loss: 1.4517
Test Accuracy: 31.60%
[INFO] Model saved.
```

Dostignut je i rezultat 36% tacnosti na test skupu

```
model_v5_multimodal x
:
Epoch 11/30
63/63 19s 295ms/step - accuracy: 0.4229 - loss: 1.3461 - val_accuracy: 0.3386 - val_loss: 1.4870 - learning_rate: 0.0010
Epoch 12/30
63/63 18s 287ms/step - accuracy: 0.4513 - loss: 1.2910 - val_accuracy: 0.3307 - val_loss: 1.4951 - learning_rate: 5.0000e-04
Epoch 13/30
63/63 18s 288ms/step - accuracy: 0.4551 - loss: 1.2912 - val_accuracy: 0.3586 - val_loss: 1.5113 - learning_rate: 5.0000e-04
Epoch 14/30
63/63 20s 311ms/step - accuracy: 0.4417 - loss: 1.2915 - val_accuracy: 0.3386 - val_loss: 1.5137 - learning_rate: 5.0000e-04
Epoch 15/30
63/63 20s 311ms/step - accuracy: 0.4471 - loss: 1.3171 - val_accuracy: 0.3665 - val_loss: 1.5111 - learning_rate: 5.0000e-04
=====
FINAL TEST EVALUATION
=====
8/8 1s 61ms/step - accuracy: 0.3735 - loss: 1.4296
Test Accuracy: 36.00%
[INFO] Model saved.

Process finished with exit code 0
```

8. Zakljucak

Projekat je komplikovan, i mozemo videti da kakvi god bili podaci, jako je tesko prediktovati kada ce ljubimac biti usvojen, jer ocigledno podaci zavise i od drugih faktora koji nisu dati ili ih je jako tesko tacno odrediti. Sentimentalnost pojedinaca ka pojedinom ljubimcu je jako tesko izraziti.

Svi modeli su u okviru 30-40% tacnosti, sa najvise dostignutom tacnoscu od ~38.5%.

Tabularni model daje solidan pocetak za rad, ali se vidi da tabularni deo nije dovoljan da precizno odredi tacnu adoption speed klasu.

Dodavanje slika preko CNN u modelima 2-5 donosi blago poboljsanje, ali ne preterano veliko, sto znaci da su tabularne osobine i dalje glavni nosioci informacija, a slike doprinose ograniceno ako se model trenira od nule na malom skupu.

Najbolji modeli su 2 i 3, cak je i 5 pokazivao dobre rezultate, multimodalni pristupi.

Verzije sa grayscale slikama (model 4 i 5) pokazuju da sam oblik i tekstura ljubimaca nose veci deo informacija o slici, a boja i razlika u osvetljenju nisu presudan faktor.

Licno, u daljem radu ovaj projekat bi se mogao proširiti:

Gledao bih da li se neki featuri mogu grupisati, pa da li zajedno nose neki bitniji faktor, vise nego kada se nalaze odvojeni. Npr ako iskoristimo breed 1 i breed 2 da vidimo da li je ljubimac mesanac ili ne. Ovo bi mozda malo doprinelo modelu.

Najveći problem jesu unbalanced klase, pogotovo veliki broj primera za klase 2 i 3, i premali broj primera za klasu 0. Ovo je veliki problem, dosta bi doprinelo kada bi imali drugaciji dataset sa vise podataka da model nauči kada treba da prediktuje klasu 0.

Dodavanje nekih optimizacija, na primer doslo mi je da bi mozda mogli da proverimo na pocetku, pomocu nekog binarnog modela, da li se očekuje da će ljubimac uopste biti usvojen ili ne. Takodje, zbog nedostataka podataka za klasu 0, da proverimo da li se uopste očekuje da će ljubimac biti usvojen u prvih 7 dana, a ako da, da li bi mozda bio usvojen istog dana. Ovo je samo ideja, ali može se razmatrati u buducnosti, jer zvuči obecavajuće.