SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Računarstvo usluga i analiza podataka

SEMINARSKI RAD

Desktop aplikacija za predviđanje vrsta životinja

Lorena Čeč

Osijek, 2024.

Sadržaj

[1. Uvod 1](#_Toc159190359)

[2. Opis problema 2](#_Toc159190360)

[2.1. Korišteni podaci 2](#_Toc159190361)

[2.2. Korišteni postupci strojnog učenja 3](#_Toc159190362)

[2.2.1. Šuma odluke 4](#_Toc159190363)

[2.2.2. Pojačano stablo odluke 5](#_Toc159190364)

[2.2.3. Logistička regresija 5](#_Toc159190365)

[3. Opis programskog rješenja 6](#_Toc159190366)

[3.1. Model strojnog učenja 6](#_Toc159190367)

[3.2. Način korištenja API-ja 8](#_Toc159190368)

[3.3. Klijentska aplikacija 8](#_Toc159190369)

[4. Zaključak 12](#_Toc159190370)

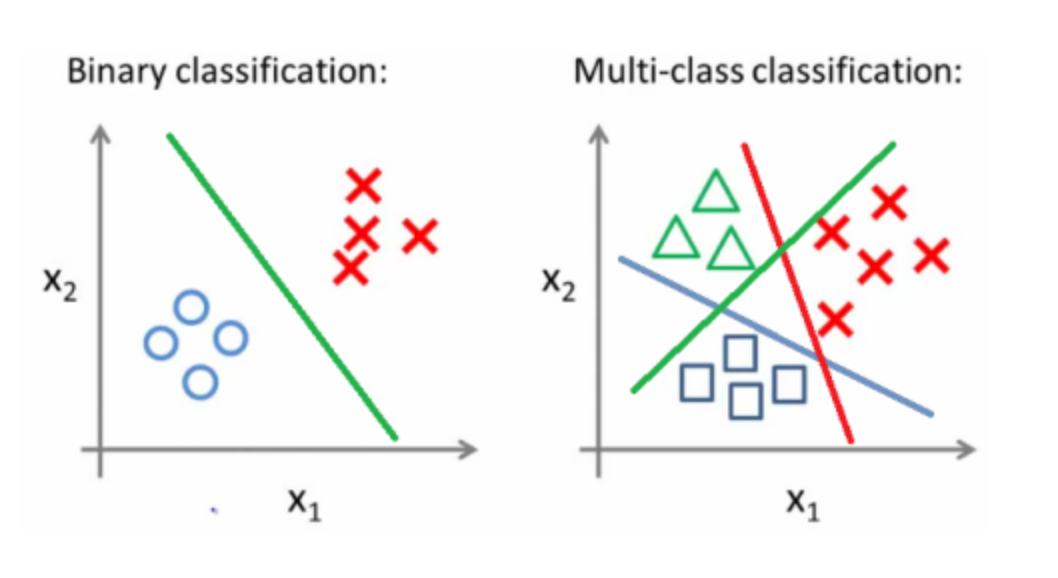
[5. Poveznice i literatura 13](#_Toc159190371)

# Uvod

Zadatak projekta je napraviti program koji će prepoznati vrstu životinje tako što se odrede specifične karakteristike životinje. Projekt sadržava aplikaciju izrađenu u Windows Forms predlošku unutar okruženja Visual Studio. U aplikaciju se unose određeni atributi koji opisuju životinje, a zatim na osnovi podataka pojavljuje se ispis vrste životinje. Aplikacija je spojena s Azure studijom za strojno učenje gdje je istreniran model na tri algoritma za višeklasnu klasifikaciju.

# Opis problema

Problem koji se javlja je problem višeklasne klasifikacije (engl. *Multi-class classification*), odnosno klasificiranja instanci u jednu od tri ili više klasa. Vizualni primjer razlike između višeklasne i binarne klasifikacije prikazan je na slici 2.1. Za razliku od binarne klasifikacije, višeklasna klasifikacija nema pojam normalnih i abnormalnih ishoda. Umjesto toga, primjeri su klasificirani kao pripadnici jedne od niza poznatih klasa. Konkretno u ovom projektu klasificiraju se instance u jednu od sedam klasa.

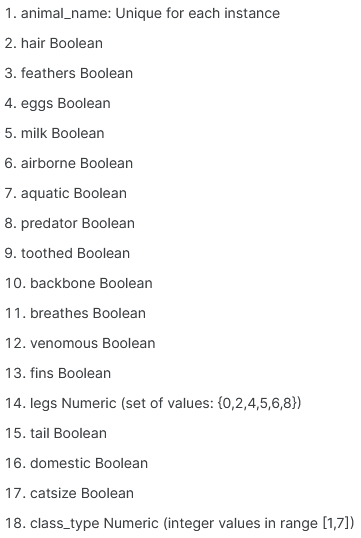
  
Slika 2.1. Razlika višeklasne i binarne klasifikacije

Načini rješavanja navedenog problema mogu se rješiti na sličan način kao kod binarnog klasificiranja jer mnogi algoritmi koji se koriste za binarnu klasifikaciju mogu se koristiti za višeklasnu klasifikaciju. Popularni algoritmi koji se mogu koristiti za višeklasnu klasifikaciju uključuju: k-Nearest Neighbors, Decision Trees, Naive Bayes, Random Forest, Gradient Boosting i sl. Algoritmi koji će biti korišteni u ovom projektu detaljnije su opisani u poglavlju 2.2. Korišteni postupci strojnog učenja.

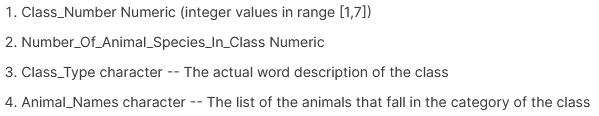
## Korišteni podaci

Korišteni podaci preuzeti su s Kaggle web stranice koja sadrži velik broj open source setova podataka.[1] Set podataka za prepoznavanje životinja sadrži nešto više od 100 unosa različitih životinja formiranih u .csv (*Comma Separated Values*) datoteku. Životinje su podjeljene na 7 vrsta: sisavac, ptica, gmaz, riba, vodozemac, kukac i beskralješnjak. Svaka životinja određena je atributima poput dlake, zubi, broju nogu, proizvodi li mlijeko i slično. Svi atributi prikazani su na slici 2.2.

Na kraju neće biti korišteni svi atributi kako bi se smanjio broj podataka koji korisnici moraju unijeti. To će se postići filtriranjem podataka na najmanji mogući broj atributa za koji se može dobiti 100%-tna preciznost pretpostavke vrste životinje.

  
Slika 2.2. Atributi prvog seta podataka

Također, koristi se još jedan set podataka koji se sastoji od svega nekoliko atributa. Ovaj set podataka sadrži ime vrste životinje, koliko se životinja iz prvog seta podatka nalazi u pojedinim vrstama te popis životinja po vrstama što je prikazano na slici 2.3. Iz ovog seta podataka saznajemo da se u prvom setu podatka nalazi ukupno 41 sisavac, 20 ptica, 5 gmazova, 13 riba, 4 vodozemca, 8 kukaca i 10 beskralježnjaka.

  
Slika 2.3. Atributi drugog seta podataka

## Korišteni postupci strojnog učenja

Nakon pripreme podataka potrebno je izabrati algoritam strojnog učenja kojima najveću stopu točnosti za pripremljeni skup podataka. Kod modela strojnog učenja jedan dio podataka odlazi na treniranje samog modela, a drugi dio na testiranje tog istreniranog modela i predviđanje točnosti na temelju ulaznih podataka. Algoritam strojnog učenja izabire se na temelju podataka s kojim se raspolaže. Algoritmi koji će se razmatrati za treniranje modela strojnog učenja su:

1. Šuma odluke (engl. *Multiclass Decision Forest*)
2. Pojačano stablo odluke (engl. *Multiclass* *Boosted* *Decision Tree*)
3. Logistička regresija (engl. *Multiclass* *Logistic regression*)

Odabir određenog algoritma izvršit će se na temelju točnosti predviđanja kod treniranja modela.

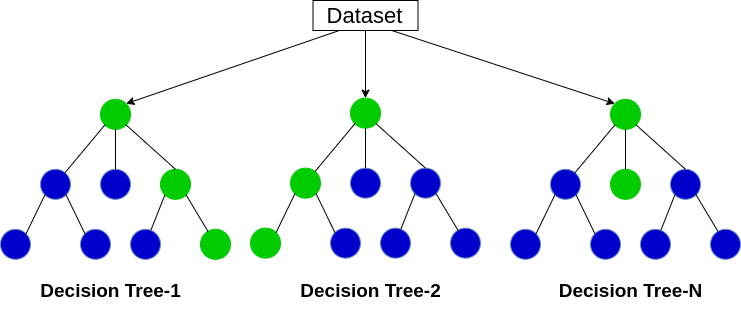
### Šuma odluke

Algoritam šume odluke skupna je metoda učenja za klasifikaciju. Algoritam radi tako što gradi više stabala odlučivanja i zatim glasa o najpopularnijoj izlaznoj klasi. Stabla odlučivanja općenito su neparametarski modeli, što znači da podržavaju podatke s različitim distribucijama. U svakom stablu izvodi se slijed jednostavnih testova za svaku klasu, povećavajući razine strukture stabla dok se ne dosegne odluka.

Stabla odlučivanja imaju mnoge prednosti:

* oni mogu predstavljati nelinearne granice odlučivanja
* učinkoviti su u računanju i korištenju memorije tijekom treninga i predviđanja
* oni izvode integrirani odabir i klasifikaciju značajki
* otporna su u prisutnosti bučnih značajki

Šuma odluke u Azure Machine Learningu sastoji se od skupa stabala odlučivanja, a općenito vrijedi da skupni modeli pružaju bolju pokrivenost i točnost od pojedinačnih stabala odlučivanja.[2]

  
Slika 2.4. Prikaz rada šume odluke

### Pojačano stablo odluke

Pojačano stablo odlučivanja metoda je skupnog učenja u kojoj drugo stablo ispravlja pogreške prvog stabla, treće stablo ispravlja pogreške prvog i drugog stabla, i tako dalje. Predviđanja se temelje na cijelom skupu stabala zajedno koji čine predviđanje. Općenito, kada su ispravno konfigurirana, pojačana stabla odlučivanja najlakše su metode pomoću kojih možete postići vrhunsku izvedbu na širokom spektru zadataka strojnog učenja. Međutim, oni su također jedni od onih koji uče više memorije, a trenutna implementacija drži sve u memoriji. Stoga, pojačani model stabla odlučivanja možda neće moći obraditi velike skupove podataka s kojima neki linearni učenici mogu rukovati.[3]

### Logistička regresija

Višeklasna logistička regresija je metoda klasifikacije koja generalizira logističku regresiju na višeklasne probleme, odnosno one s više od dvije moguće diskretne izlazne vrijednosti. Drugim riječima, to je model koji se koristi za predviđanje vjerojatnosti različitih mogućih ishoda kategorički distribuirane zavisne varijable, uz zadani skup neovisnih varijabli (koje mogu biti stvarne, binarne, kategoričke, itd.).[4]

# Opis programskog rješenja

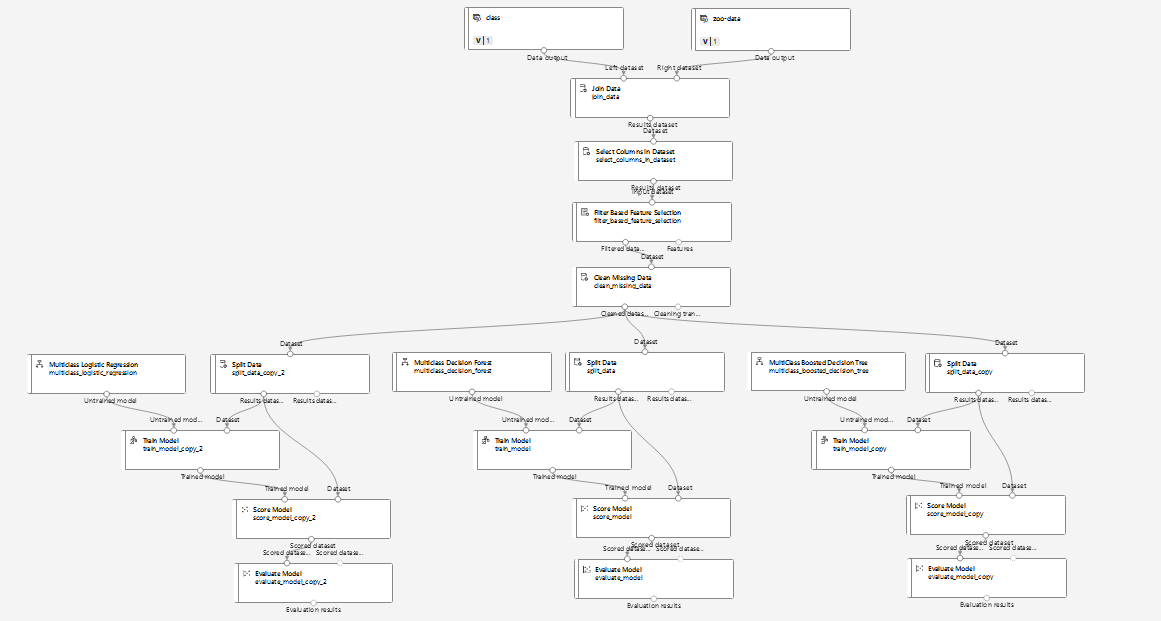
Programsko rješenje zadatka sastoji se od sljedećih dijelova:

1. Model strojnog učenja
2. API koji povezuje model strojnog učenja i korisničko sučelje
3. Korisničko sučelje izvedeno kao desktop aplikacija

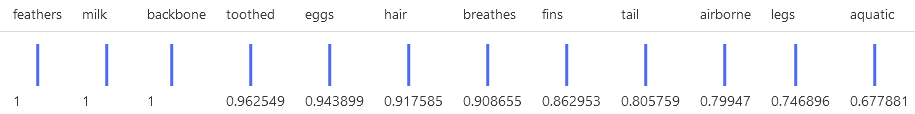
## Model strojnog učenja

Model strojnog učenja izrađen je u Azure Machine Learning Studio platformi. Izrađen je po uzoru na model s laboratorijskih vježbi iz kolegija Računarstvo usluga i analiza podataka. Odlučeno je uzeti 70% podataka za trening, a ostalih 30% za test. Nakon raspodijele podataka potrebno je izabrati algoritam strojnog učenja. Algoritam izabriremo između tri već navedena algoritma: binarna šuma, stablo odluke i logistička regresija.

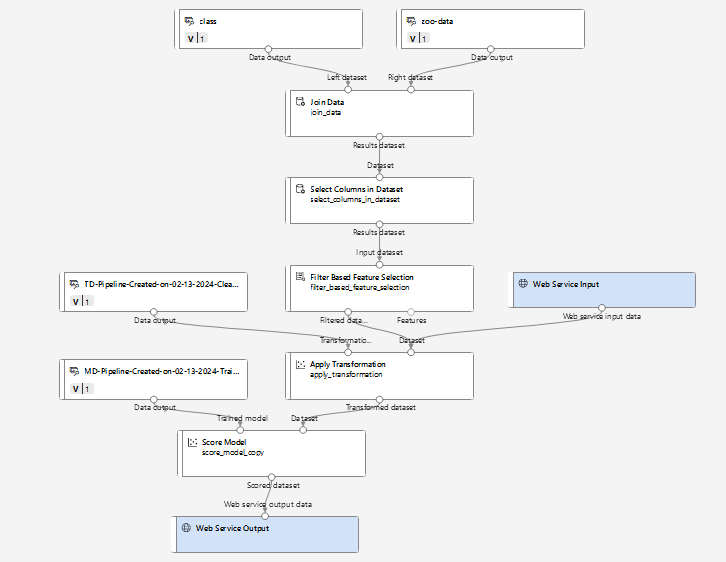
Slika 3.1. prikazuje raspodjelu na tri algoritma na osnovu koje ćemo odabrati najefikasniji algoritam.

  
Slika 3.1. Treniranje modela strojnog učenja

Kao što je prikazano na slici 3.1. imamo 2 CSV seta podataka koje spajamo blokom *Join Data*. Potom blokom *Select Columns in Dataset* izbacujemo neke od stupaca koji nam nisu potrebni (stupci poput rednog broja vrste životinja). Blokom *Filter Based Feature Selection* izvlačimo optimalan broj atributa pomoću kojih možemo točno odrediti vrstu životinje. Pomoću navedenog programskog elementa dolazimo do zaključka da nam je potrebno 12 od sveukupno 17 atributa iz seta podataka kako bi odredili klasu uz stopostotnu preciznost. Prikaz ovih 12 atributa prikazan je na slici 3.2. Blokom *Clean Missing Data* spriječujemo probleme uzrokovane nedostatkom podataka koji mogu nastati prilikom obuke modela.

  
Slika 3.2. Prikaz korištenih atributa

Za svaki algoritam koristimo blok Split data, Train model, Score model i Evaluate model. U slučaju treniranja modela s 12 atributa svi algoritmi su imali 100%-tnu preciznost, no pri treniranju modela gdje se koristilo manje od 12 atributa najbolji se pokazao algoritam stablo odluke (engl. Multiclass Boosted Decision Tree) pa je stoga ovaj algoritam korišten.

  
Slika 3.3. Inference pipeline

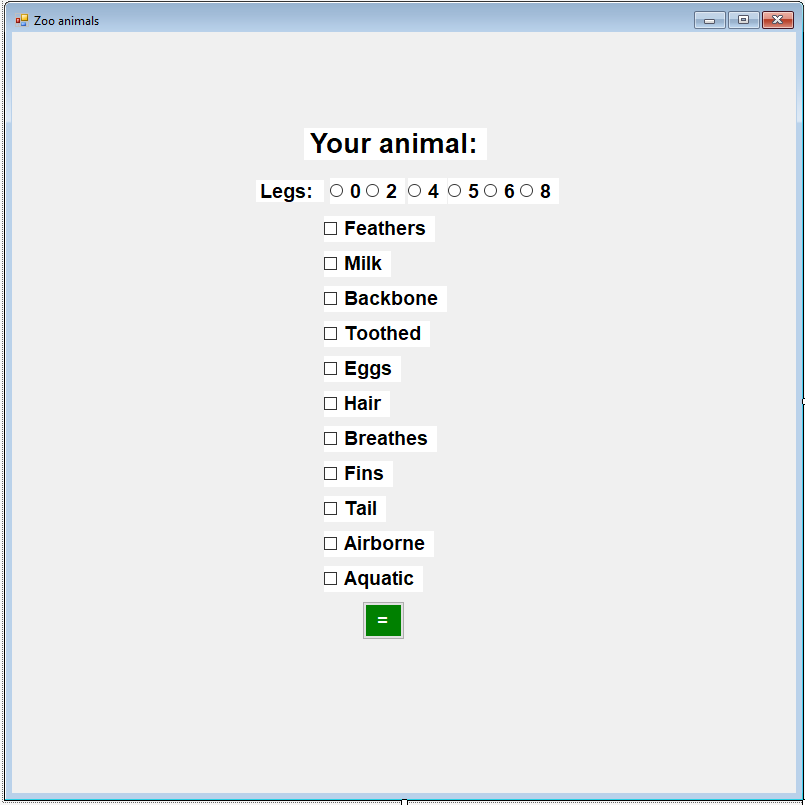
Na slici 3.3. prikazan je Inference pipeline kod kojeg je dodan blok Web Service Input kako bi se mogli poslati podaci modelu. Na osnovu unosa podataka od strane korisnika, istrenirani model vrši predviđanje nakon čega vraća rezultat predviđanja.

## Način korištenja API-ja

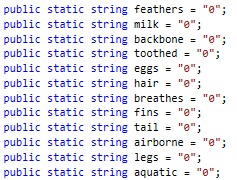
Nakon *deploymenta* modela, generiran je API za korištenje modela unutar neke korisničke aplikacije. U ovom projektu korišten je generirani kod za programski jezik C#. Kod je minimalno izmijenjen kako bi bilo lakše rukovati njime što je dodatno objašnjeno u sljedećem poglavlju gdje je vidljiv cijeli kod aplikacije.

## Klijentska aplikacija

Za izradu aplikacije korišten je Windows Forms predložak u Visual Studiju. Na slici 3.4. prikazan je prvotni oblik predloška u kojem su izlistani glavnih 12 atributa ključnih za prepoznavanje vrste životinja. Na osnovi ovog predloška dobivamo potrebne informacije koje će biti poslane na model strojnog učenja.

  
Slika 3.4. Prvotni izgled aplikacije

Programski kod 3.1. prikazuje inicijalizaciju atributa sa slike 3.3. gdje su sve vrijednosti postavljene na nulu.

  
Programski kod 3.1. Inicijalizacija atributa

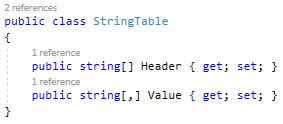
Programski kod 3.2. prikazuje funkciju ButtonClick koja ukoliko je atribut sa slike 3.3. označen kvačicom, mijenja se vrijednost atributa iz nule u jedinicu. U suprotnom, vrijednost atributa je nula. Jedino za atribut vezan za broj nogu odnosno atribut *legs* odvojeno se postavljaju vrijednosti u ovisnosti koji broj je odabran (0, 2, 4, 5, 6, 8). Nakon postavljanja vrijednosti svih atributa, poziva se metoda InvokeRequestResponseService() koja je definirana prilikom završetka deploymenta.

  
Programski kod 3.2. Funkcija ButtonClick

Programski kod 3.3. prikazuje postavljanje atributa dobivenih unutar aplikacije na način da je svaki atribut priložen uz njegovu inačicu u modelu strojnog učenja.

  
Programski kod 3.3. Postavljanje varijable

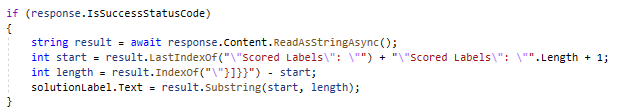
Programski kod 3.4. prikazuje klasu koja je korištena u programskom kodu 3.3. za lakše postavljanje varijable koja će se predati kao argument pri slanju podataka na model.

  
Programski kod 3.4. Postavljanje varijable

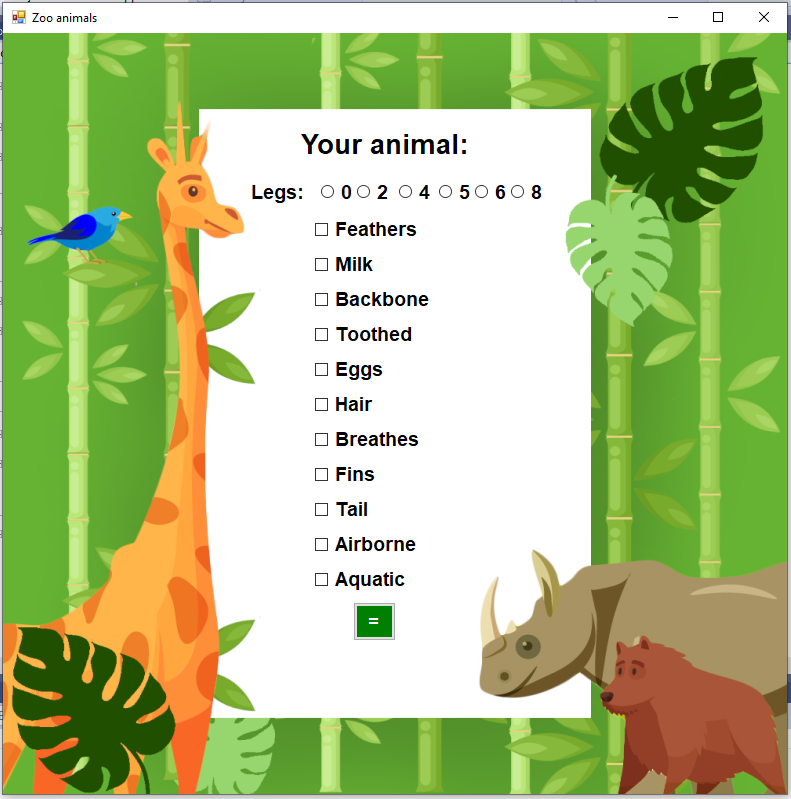
Programski kod 3.5. prikazuje predefinirani kod koji povezuje aplikaciju s modelom i šalje podatke definirane u programskom kodu 3.3.

  
Programski kod 3.5. Povezivanje aplikacije s modelom

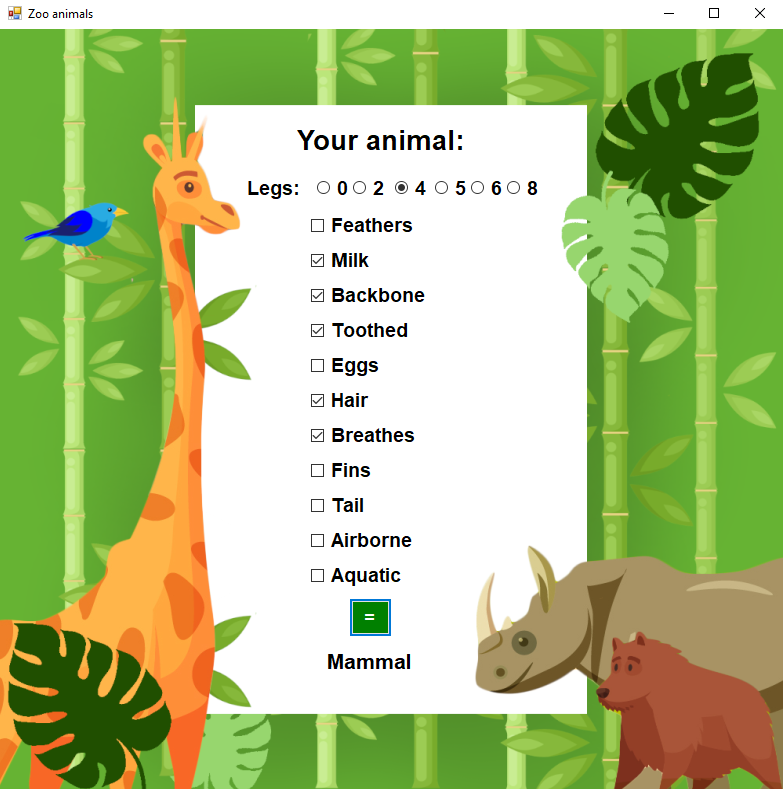
Programski kod 3.6. prikazuje dohvaćanje rezultata u obliku stringa te iz tog podatka izvlači se rezultat odnosno vrsta životinje dobivena na osnovi traženih atributa.

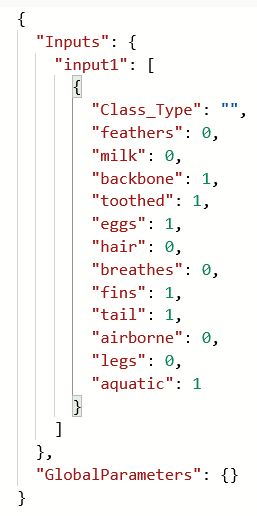
  
Programski kod 3.6. Dohvaćanje rezultata i ispis rezultata

Slika 3.5. prikazuje izgled aplikacije kada se pokrene odnosno bez unosa korisnika.

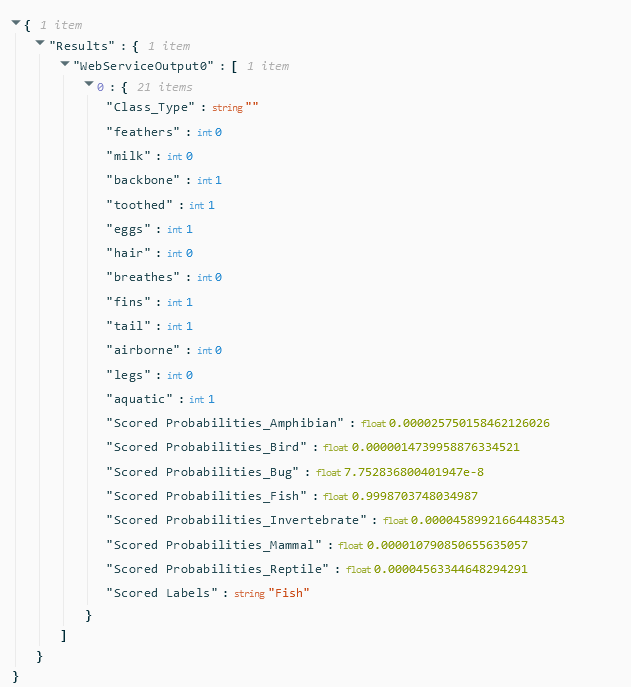
  
Slika 3.5. Izgled pokrenute aplikacije

Slika 3.6. prikazuje izgled aplikacije gdje uz unesene karakteristike životinje dobiva se rezultat o kojoj vrsti životinje se radi.

  
Slika 3.6. Izgled aplikacije uz unos podataka i rezultat

  
Slika 3.7. Prikaz testnih podataka

Na slici 3.7. prikazan je skup atributa koji se testiraju u Azure Machine Learning Studiju, dok su na slici 3.8. prikazani rezultati testa odnosno koja je vrsta životinje i postotak preciznosti.

  
Slika 3.8. Prikaz rezultata

# Zaključak

U ovom projektu koristio se Azure Machine Learning Studio gdje su se trenirala i testirala tri algoritma strojnog učenja te uspješno je stvoren model za predviđanje vrste životinje. Odabrani algoritam pomoću unesenih karakteristika životinje svrstava navedenu životinju u jednu od sedam vrsta: sisavac, ptica, gmaz, riba, vodozemac, kukac i beskralješnjak. Izrađeni model i aplikacija stvorena u Visual Studio okruženju pružaju programsko rješenje problema višeklasne klasifikacije.

U radu se težilo pojednostavljenju korištenog seta podataka te stoga se na kraju koristio manji broj podataka potrebnih za unos. Također, Azure Machine Learning Studio se kroz rad pokazao kao kvalitetno razvojno okruženje za rad sa strojnim učenjem.

# Poveznice i literatura

Programskom je rješenju moguće pristupiti preko:

|  |
| --- |
| [Programsko rješenje na GitHubu](https://github.com/Lorena-Cec/Zoo-classification-project) |

1. Kaggle, Zoo Animal Classification, <https://www.kaggle.com/datasets/uciml/zoo-animal-classification>
2. Microsoft Azure, Multiclass Decision Forest, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-forest>
3. Microsoft Azure, Two-Class Boosted Decision Tree, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-boosted-decision-tree>
4. Wikipedia, Multinomial Logistic Regression, https://en.wikipedia.org/wiki/Multinomial\_logistic\_regression