Animal-inator

Katie Prescott, Zhiyuan (James) Zhang, Brandon Rothweiler, Luke Smith

Approach to the game

- Inspired by Akinator the Genie, which classifies characters
- We decided to classify animals

About the dataset

- Animals with Attributes 2 (AwA2) (https://cvml.ist.ac.at/AwA2/)
- 50 animal classes with 85 attributes each (pre-extracted features)
  - Binary data set: Features rated 0 or 1
  - Continuous data set: Features rated 0-100
- 37,322 images total


antelope
grizzly bear
killer whale
beaver
dalmatian
persian cat
horse
german shepherd
blue whale
siamese cat
skunk
mole
tiger
hippopotamus
leopard
moose
spider monkey
humpback whale
elephant
gorilla
ox
fox
sheep
seal
chimpanzee
hamster
squirrel
rhinoceros
rabbit
bat
giraffe
wolf
chihuahua
rat
weasel
otter
buffalo
zebra
giant panda
deer
bobcat
pig
lion
mouse
polar bear
collie
walrus
raccoon
cow
dolphin
black
white
blue
brown
gray
orange
red
yellow
patches
spots
stripes
furry
hairless
toughskin
big
small
bulbous
lean
flippers
hands
hooves
pads
paws
longleg
longneck
tail
chewteeth
meatteeth
buckteeth
strainteeth
horns
claws
tusks
smelly
flys
hops
swims
tunnels
walks
fast
slow
strong
weak
muscle
bipedal
quadrapedal
active
inactive
nocturnal
hibernate
agility
fish
meat
plankton
vegetation
insects
forager
grazer
hunter
scavenger
skimmer
stalker
newworld
oldworld
arctic
coastal
desert
bush
plains
forest
fields
jungle
mountains
ocean
ground
water
tree
cave
fierce
timid
smart
group
solitary
nestspot
domestic

Animals

antelope, grizzly bear, killer whale, beaver, dalmatian, persian cat, horse, german shepherd, blue whale, siamese cat, skunk, mole, tiger, hippopotamus, leopard, moose, spider monkey, humpback whale, elephant, gorilla, ox, fox, sheep, seal, chimpanzee, hamster, squirrel, rhinoceros, rabbit, bat, giraffe, wolf, chihuahua, rat, weasel, otter, buffalo, zebra, giant panda, deer, bobcat, pig, lion, mouse, polar bear, collie, walrus, raccoon, cow, dolphin

Attributes

black, white, blue, brown, gray, orange, red, yellow, patches, spots, stripes, furry, hairless, toughskin, big, small, bulbous, lean, flippers, hands, hooves, pads, paws, longleg, longneck, tail, chewteeth, meatteeth, buckteeth, strainteeth, horns, claws, tusks, smelly, flys, hops, swims, tunnels, walks, fast, slow, strong, weak, muscle, bipedal, quadrapedal, active, inactive, nocturnal, hibernate, agility, fish, meat, plankton, vegetation, insects, forager, grazer, hunter, scavenger, skimmer, stalker, newworld, oldworld, arctic, coastal, desert, bush, plains, forest, fields, jungle, mountains, ocean, ground, water, tree, cave, fierce, timid, smart, group, solitary, nestspot, domestic

Initial Idea

- Using the continuous dataset.
- The player picks an animal.
- We ask the player about their animal's attributes using the specified features on a scale of 0 to 100.
- Use KNN to pick the nearest neighbors to their features and see if their chosen animal is there!

The Problem with KNN

- Using the binary dataset:
  - Every time the program runs, player need to answer all 85 questions.
- Using the continuous dataset:
  - Everyone has different views on the value of a feature on the scale of 0 --- 100.
- The error rate of using KNN could potentially be very high.

Our Second Approach

- Using a decision tree with Binary dataset.
- At each node of the tree, ask the user if their animal meets the splitting criteria
  - If it does, go down one side of the tree
  - If it doesn't, go down the other side of the tree
- Keep asking until only a single animal remains.

Splitting Criterion

Minimum Entropy

- Tree will be balanced, always using the most optimal splits
- Game asks the same questions every time in the same order -> Less interesting gameplay

Gini Impurity

- Tree is not the most balanced possible tree
- Different questions every time the game is played -> More interesting gameplay

1. $Gini: Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$
2. $Entropy: H(E) = -\sum_{j=1}^{c} p_j \log p_j$

Clustering

- After the game is over, we wanted to show the user a list of animals similar to their animal
- We use 12 clusters to determine animals that are similar to each other
- Our clustering algorithm uses regular euclidean distance

Animal Clusters

- **Primates (+Bat)**: {spider monkey, gorilla, chimpanzee, bat}
- **Dirty/Smelly Animals**: {skunk, mole, hamster, squirrel, rabbit, mouse}
- **Sea Creatures**: {killer whale, blue whale, humpback whale, seal, walrus, dolphin}
- **Farm Animals (+Moose)**: {moose, ox, sheep, buffalo, pig, cow}
- **Predators**: {tiger, leopard, fox, wolf, bobcat, lion}
- **Prairie Animals**: {antelope, horse, giraffe, zebra, deer}
- **Dogs & Cats**: {dalmatian, persian cat, german shepherd, siamese cat, chihuahua, collie}
- **Bears**: {grizzly bear, polar bear}
- **Large Gray Animals**: {hippopotamus, elephant, rhinoceros}
- **Semiaquatic**: {beaver, otter}
- **Rodent-Like**: {rat, weasel, raccoon}
- **Giant Panda**: {giant panda}

KNN Strikes Back

We wanted to give KNN another chance, so we created an alternative version of the game:

- User picks 5 animals that are most similar to their animal
- Program runs KNN algorithm 5 times (with k = 5, generating 25 total neighbors), but excluding the animals that the user supplied
- Program outputs the animal that appears most frequently in the neighbor lists

Design Decisions

- We decided to use Python because it allows for easy I/O and data structure manipulation
- Python also provides the scikit library which provides functions for creating decision trees and clustering data

Demo

Future Improvements

- Backtracking through the tree - not require all user answers to be correct
- Ask additional questions at the end so that the game can confirm its guess is correct
- Some animals may be unreachable for second game