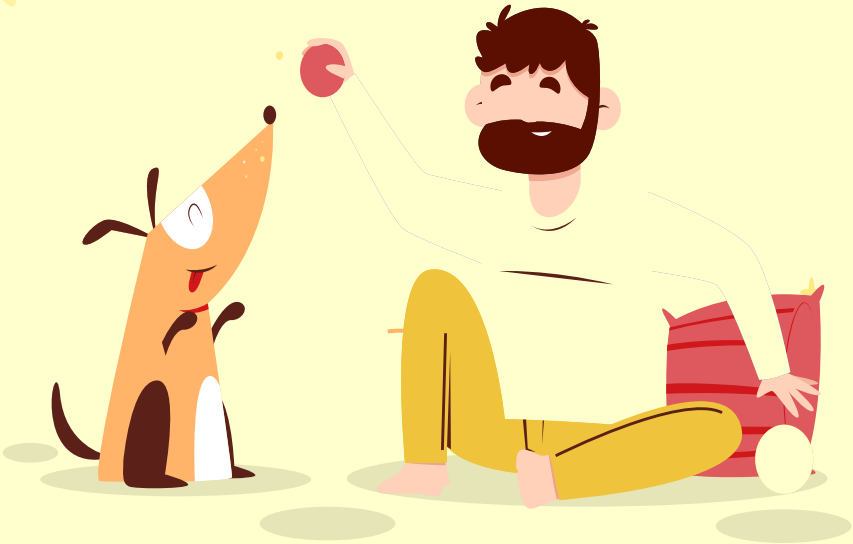
Dog Recommender System

P2 Group 1
1901851 Crystal Toh Yi Shan
1901888 Teng Zheng Yu
1901854 The Nu Win
1901828 Wu Jia Jie

TABLE OF CONTENTS

01
Problem Statement



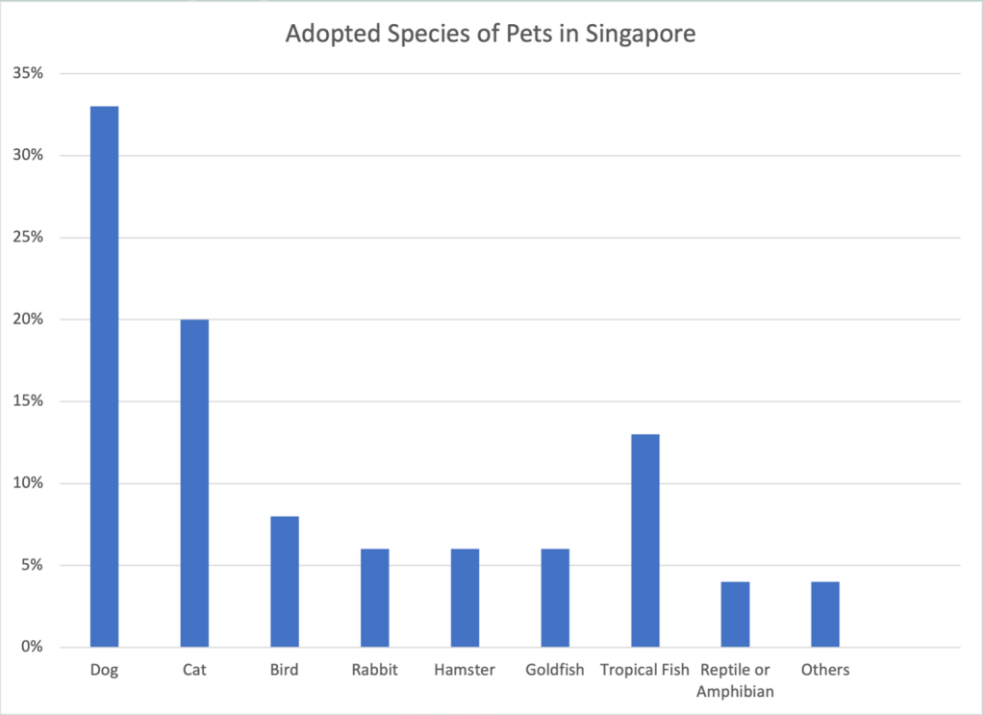
THE STRAITS TIMES LIFE

More people in Singapore interested in adopting or fostering pets during Covid-19 pandemic

Undergraduate Sarah Chua and her family adopted Tau Pok during the circuit breaker period. PHOTO: COURTESY OF SARAH CHUA

https://www.straitstimes.com/lifestyle/home-design/more-interested-in-adopting-or-fostering-pets-during-covid-19-pandemic-as-they

Online Survey Results Conducted in 2021



Pet Dog was ranked 1st

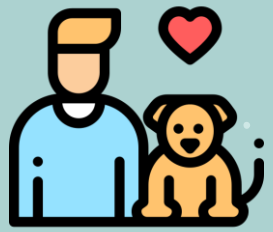
as the most adopted pet in Singapore

https://insight.rakuten.com/pet-ownership-in-asia/

Reasons for Adopting Pets

41%

To feel less stressed



36%

To have some company

36%

To feel more secure

26%

To be more physically active

https://insight.rakuten.com/pet-ownership-in-asia/

Limitations of Existing Dog Recommender Systems

- Only recommend one breed to user
- Not based on specific context of singapore living conditions

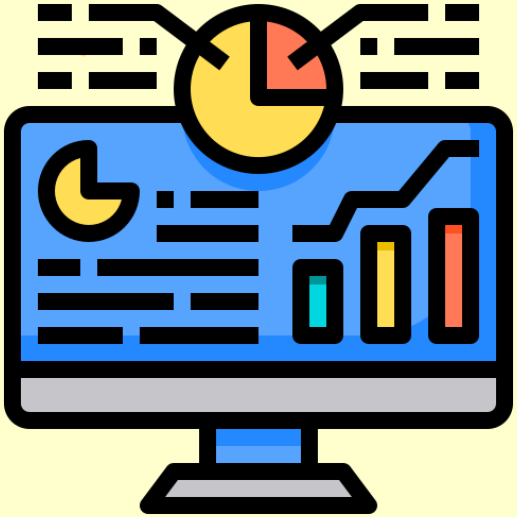
  (eg. Hot weather instead of cold, close living quarters within HDB)

Proposed Solution

- Based on user's desired traits in dogs
- Recommend top 5 breeds to consider

02

Data Preprocessing



Data Preprocessing Process

1) Data Preparation

Identifying and correcting mistakes

2) Feature Selection

Correlation

Eliminate features using similarity or similarities

3) Dimensionality Reduction

PCA

Introducing Dataset

```
# load the dataset
dataset = pd.read_csv('dogs.csv', na_values='?', index_col='|')
print(dataset.shape)
```

replacing NA values with ? and defining the y_labels 'dog breeds' as the index columns

```
<class 'pandas.core.frame.DataFrame'>
Index: 199 entries, Azawakh to Leonberger
Data columns (total 40 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   url                     199 non-null    object
 1   shedding                199 non-null    int64
 2   overall_health          199 non-null    int64
 3   groom                   199 non-null    int64
 4   weight_gain             198 non-null    float64
 5   drooling                199 non-null    int64
 6   general_health          198 non-null    float64
 7   size                    198 non-null    float64
 8   wander                  198 non-null    float64
 9   intelligence            199 non-null    int64
 10  overall_trainability    199 non-null    int64
 11  prey_drive              194 non-null    float64
 12  mouthiness              198 non-null    float64
 13  bark                    198 non-null    float64
 14  train                   198 non-null    float64
 15  playful                 198 non-null    float64
 16  energy                  199 non-null    int64
 17  exercise                199 non-null    int64
 18  overall_exerciseneeds   199 non-null    int64
 19  exercise_intensity      198 non-null    float64
 20  cold_weather            199 non-null    int64
 21  novice_owners           199 non-null    int64
 22  sensitivity             199 non-null    int64
 23  overall_adaptability    199 non-null    int64
 24  hot_weather             199 non-null    int64
 25  alone                   199 non-null    int64
 26  apartment               199 non-null    int64
 27  family_affection        199 non-null    int64
 28  friendly_strangers      199 non-null    int64
 29  overall_friendly        199 non-null    int64
 30  kid_friendly            199 non-null    int64
 31  dog_friendly            199 non-null    int64
 32  breed_group             199 non-null    object
 33  max_lifespan            199 non-null    int64
 34  min_lifespan            199 non-null    object
 35  max_weight              193 non-null    float64
 36  min_weight              193 non-null    object
 37  min_height              198 non-null    object
 38  max_height              159 non-null    object
 39  shoulder_height         117 non-null    object
dtypes: float64(11), int64(22), object(7)
memory usage: 63.7+ KB
```

Data Preparation

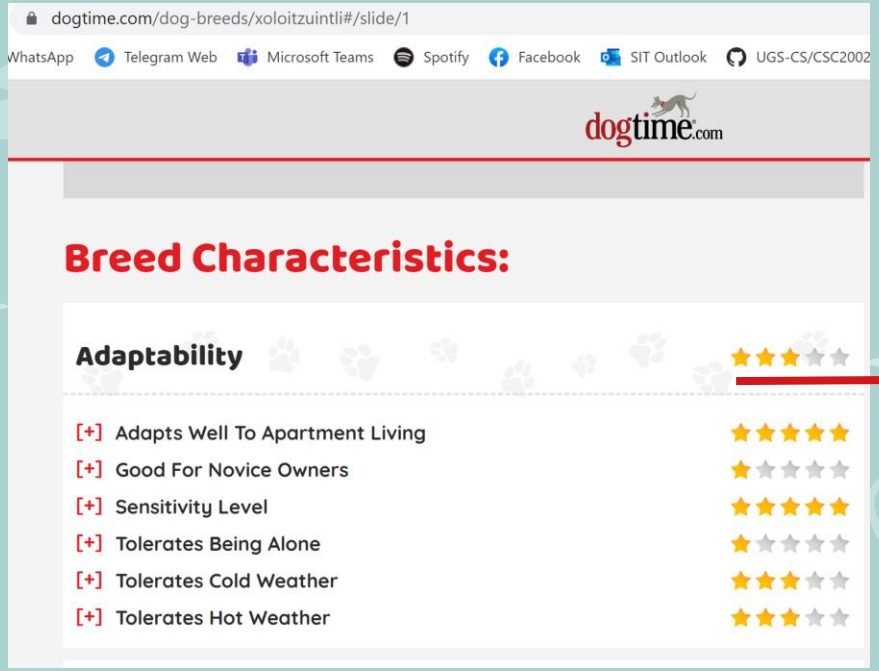
The shape of our dataset is given as (199,40)

observations

1. **Irrelevant features. Eg: url**
2. **Inconsistent data type**
3. **Missing data**
4. **Repetitive data (?)**

Data Preparation - Irrelevant

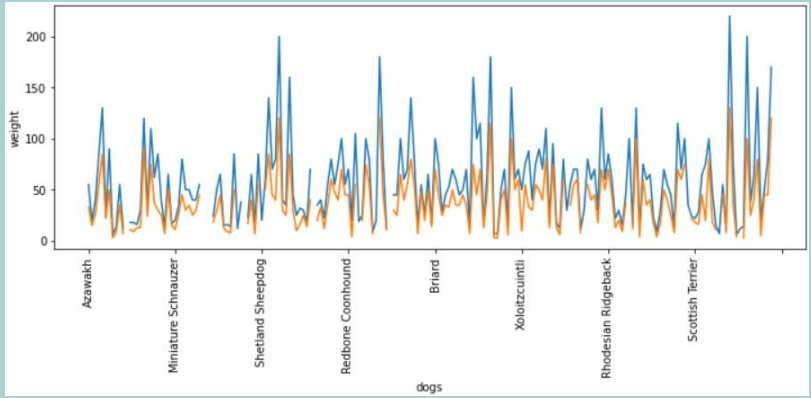
Overall categories does not contain meaningful insights.



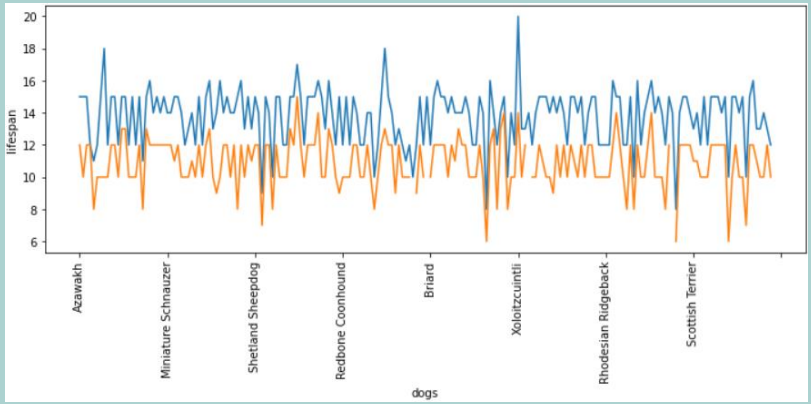
overall categories are dropped

Irrelevant features

Max & Min weight



Max & Min lifespan



From the table above, the maximum and minimum of each feature is similar. The range of weight and lifespan are not useful for analysis since mean ! = range/2.

Irrelevant features

Other irrelevant features includes:

1. Tolerance to cold weather (not relevant in Singapore context)

```
<class 'pandas.core.frame.DataFrame'>
Index: 199 entries, Azawakh to Leonberger
Data columns (total 40 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   url                    199 non-null    object
 1   shedding               199 non-null    int64
 2   overall_health         199 non-null    int64
 3   groom                  199 non-null    int64
 4   weight_gain            198 non-null    float64
 5   drooling               199 non-null    int64
 6   general_health         198 non-null    float64
 7   size                   198 non-null    float64
 8   wander                 198 non-null    float64
 9   intelligence           199 non-null    int64
 10  overall_trainability   199 non-null    int64
 11  prey_drive             194 non-null    float64
 12  mouthiness             198 non-null    float64
 13  bark                   198 non-null    float64
 14  train                  198 non-null    float64
 15  playful                198 non-null    float64
 16  energy                 199 non-null    int64
 17  exercise               199 non-null    int64
 18  overall_exerciseneeds  199 non-null    int64
 19  exercise_intensity     198 non-null    float64
 20  cold_weather           199 non-null    int64
 21  novice_owners          199 non-null    int64
 22  sensitivity            199 non-null    int64
 23  overall_adaptability   199 non-null    int64
 24  hot_weather            199 non-null    int64
 25  alone                  199 non-null    int64
 26  apartment              199 non-null    int64
 27  family_affection       199 non-null    int64
 28  friendly_strangers     199 non-null    int64
 29  overall_friendly       199 non-null    int64
 30  kid_friendly           199 non-null    int64
 31  dog_friendly           199 non-null    int64
 32  breed_group            199 non-null    object
 33  max_lifespan           199 non-null    int64
 34  min_lifespan           199 non-null    object
 35  max_weight             193 non-null    float64
 36  min_weight             193 non-null    object
 37  min_height             198 non-null    object
 38  max_height             159 non-null    object
 39  shoulder_height        117 non-null    object
dtypes: float64(11), int64(22), object(7)
memory usage: 63.7+ KB
```

Data Preparation - Inconsistent Data Type

```
dataset['min_lifespan']= pd.to_numeric(dataset['min_lifespan'],errors='coerce')
dataset['min_weight']= pd.to_numeric(dataset['min_weight'],errors='coerce')
```

24	hot_weather	199 non-null	int64
25	alone	199 non-null	int64
26	apartment	199 non-null	int64
27	family_affection	199 non-null	int64
28	friendly_strangers	199 non-null	int64
29	overall_friendly	199 non-null	int64
30	kid_friendly	199 non-null	int64
31	dog_friendly	199 non-null	int64
32	breed_group	199 non-null	object
33	max_lifespan	199 non-null	int64
34	min_lifespan	199 non-null	object
35	max_weight	193 non-null	float64
36	min_weight	193 non-null	object
37	min_height	198 non-null	object
38	max_height	159 non-null	object
39	shoulder_height	117 non-null	object

20	alone	199 non-null	int64
21	apartment	199 non-null	int64
22	family_affection	199 non-null	int64
23	friendly_strangers	199 non-null	int64
24	kid_friendly	199 non-null	int64
25	dog_friendly	199 non-null	int64
26	breed_group	199 non-null	object
27	max_lifespan	199 non-null	int64
28	min_lifespan	195 non-null	float64
29	max_weight	193 non-null	float64
30	min_weight	183 non-null	float64

```
# format fields
dataset = dataset.astype({header[4]: 'int64', header[6]: 'int64', header[7]: 'int64', header[8]: 'int64',
                          header[11]: 'int64', header[12]: 'int64', header[13]: 'int64', header[14]: 'int64',
                          header[15]: 'int64', header[19]: 'int64', header[33]: 'int64', header[34]: float,
                          header[36]: float})
dataset.info()
```

Data Preparation - Missing data

```python
for i in dataset.columns:
    num_missing = (dataset[[i]].isnull()).sum()
    perc = num_missing/dataset.shape[0]*100
    print('> %s, Missing: %d (%.1f%%)' % (i,num_missing,perc))
```

```
> shedding, Missing: 0 (0.0%)
> groom, Missing: 0 (0.0%)
> weight_gain, Missing: 1 (0.5%)
> drooling, Missing: 0 (0.0%)
> general_health, Missing: 1 (0.5%)
> size, Missing: 1 (0.5%)
> wander, Missing: 1 (0.5%)
> intelligence, Missing: 0 (0.0%)
> prey_drive, Missing: 5 (2.5%)
> mouthiness, Missing: 1 (0.5%)
> bark, Missing: 1 (0.5%)
> train, Missing: 1 (0.5%)
> playful, Missing: 1 (0.5%)
> energy, Missing: 0 (0.0%)
> exercise, Missing: 0 (0.0%)
> exercise_intensity, Missing: 1 (0.5%)
> cold_weather, Missing: 0 (0.0%)
> novice_owners, Missing: 0 (0.0%)
> sensitivity, Missing: 0 (0.0%)
> hot_weather, Missing: 0 (0.0%)
> alone, Missing: 0 (0.0%)
> apartment, Missing: 0 (0.0%)
> family_affection, Missing: 0 (0.0%)
> friendly_strangers, Missing: 0 (0.0%)
> kid_friendly, Missing: 0 (0.0%)
> dog_friendly, Missing: 0 (0.0%)
> breed_group, Missing: 0 (0.0%)
> max_lifespan, Missing: 0 (0.0%)
> min_lifespan, Missing: 4 (2.0%)
> max_weight, Missing: 6 (3.0%)
> min_weight, Missing: 16 (8.0%)
```

```python
# inpute missing values
bool_series = pd.isnull(dataset["weight_gain"])
dataset[bool_series]
```

breed	shedding	groom	weight_gain	drooling	general_health	size	w
Korean Jindo Dog	3	4	NaN	1	NaN	NaN	

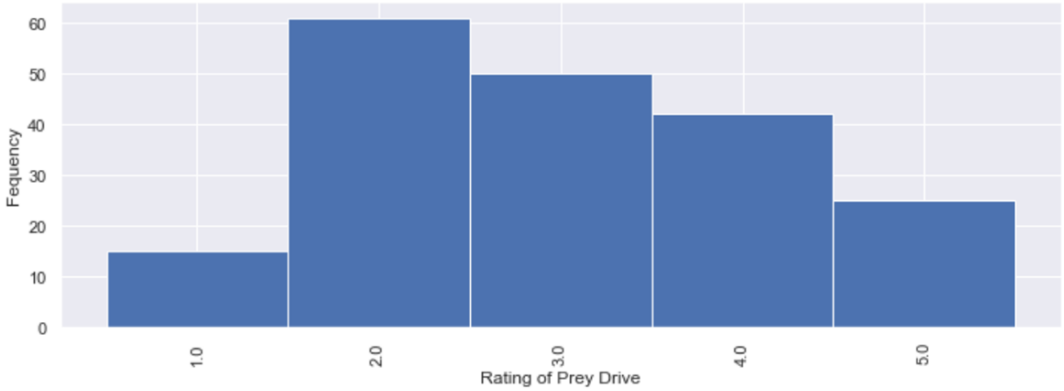
1 rows × 31 columns

This particular breed has at least 9 missing features, hence drop the breed

Data Preparation - Missing data

Next, we visualize the prey_drive attribute to determine how to handle the missing values

```
prey_drive = dataset['prey_drive'].value_counts().sort_index()
fig, ax = plt.subplots(figsize=(12, 4))
ax = prey_drive.plot(kind='bar',width=1.0)
ax.set(xlabel = "Rating of Prey Drive",
       ylabel = "Fequency")
plt.show()
print('The median of prey_drive rating is:',dataset["prey_drive"].median())
print('The mean of prey_drive rating is:',round(dataset["prey_drive"].mean(),0))
```




```
The median of prey_drive rating is: 3.0
The mean of prey_drive rating is: 3.0
```

Impute missing values with either mean or median

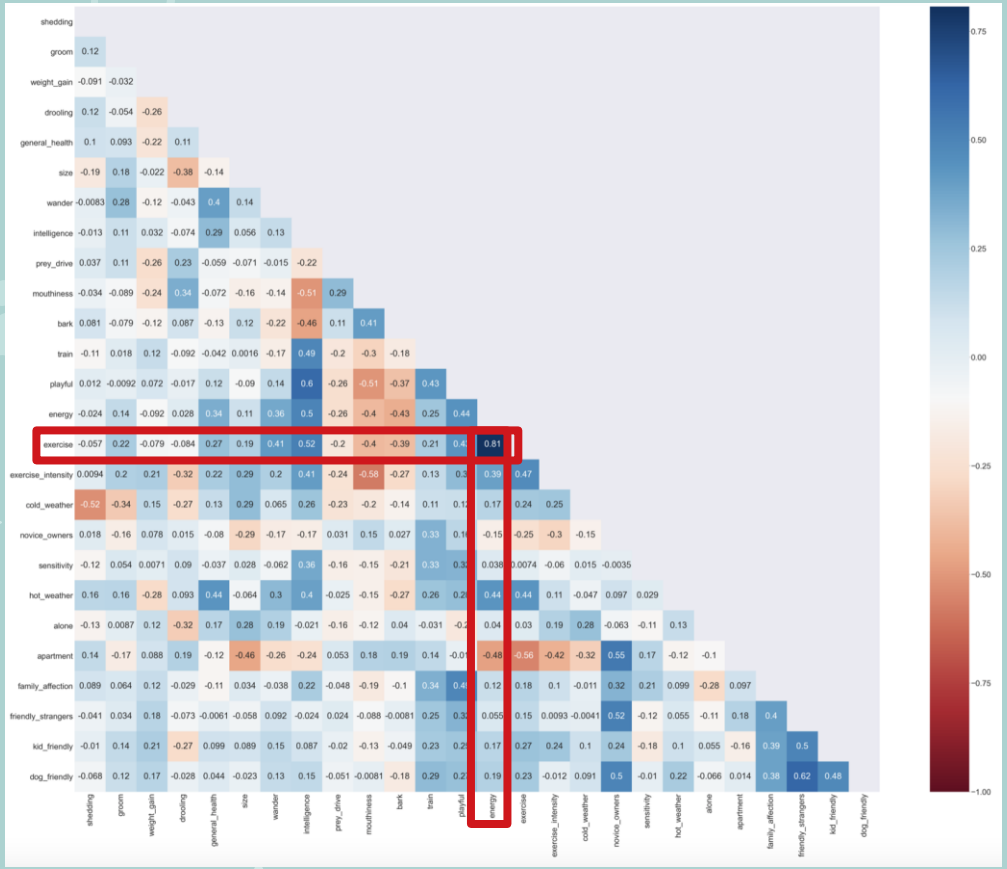
Data Preparation - Repetitive data

```
columnStatistics = pd.DataFrame(dataset.max(axis=0))
columnStatistics.columns = ['MaxValues']

columnStatistics['MinValues'] = dataset.min(axis=0)
uniqueCounts = pd.DataFrame(columnStatistics.index)
uniqueCounts.set_index(0, inplace=True)
uniqueCounts['UniqueValues'] = np.nan
for col in dataset:
    uniqueCounts.loc[col]['UniqueValues'] = dataset[col].nunique()
columnStatistics['UniqueValues'] = uniqueCounts['UniqueValues']

columnStatistics
# likert scale from a scale of 1 to 5, no zero min val
```

	MaxValues	MinValues	UniqueValues
shedding	5	1	5.0
groom	5	1	5.0
weight_gain	5.0	1.0	5.0
drooling	5	1	5.0
general_health	5.0	1.0	5.0
size	5.0	1.0	5.0
wander	5.0	1.0	5.0
intelligence	5	2	4.0
prey_drive	5.0	1.0	5.0
mouthiness	5.0	1.0	5.0
bark	5.0	1.0	5.0

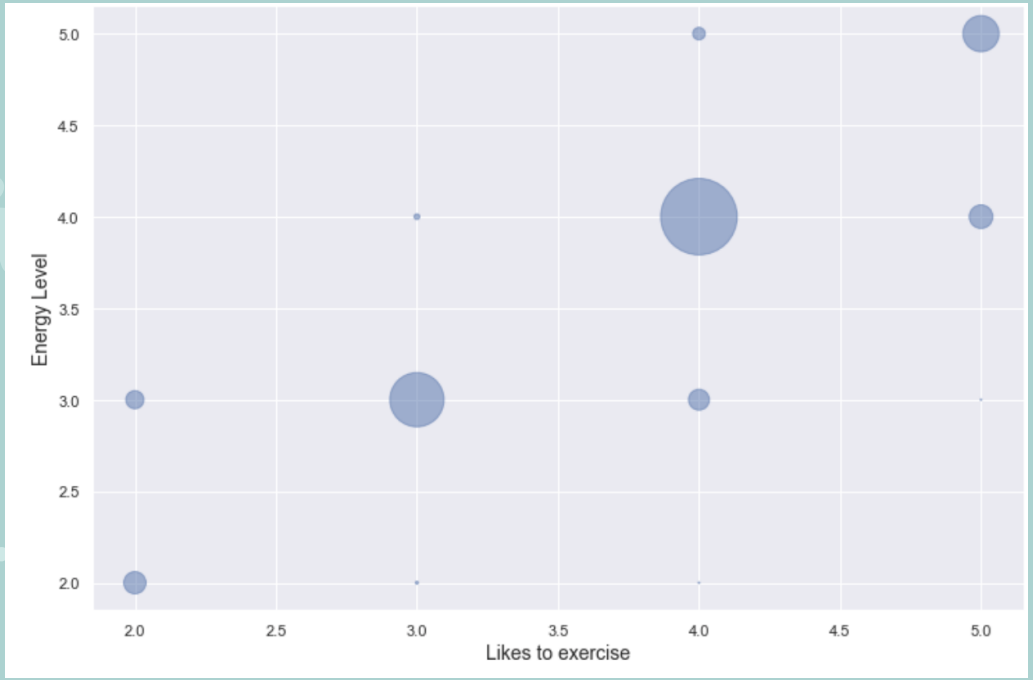
No features in this dataset is rated with a single value only

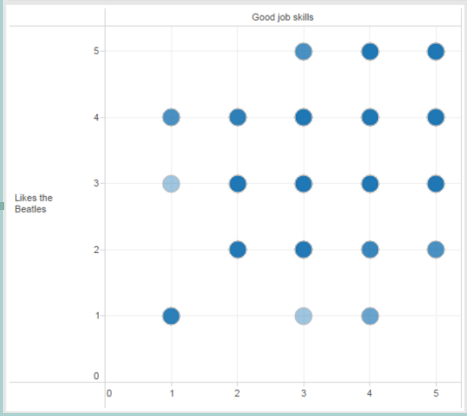
Feature Selection - Correlation



Exercise and Energy are highly correlated with the value of 0.81.

Feature Selection - Correlation





visualize

since this is a likert scale data, scatterplot is not optimal in visualizing the data hence a bubble chart is used with an additional frequency parameter

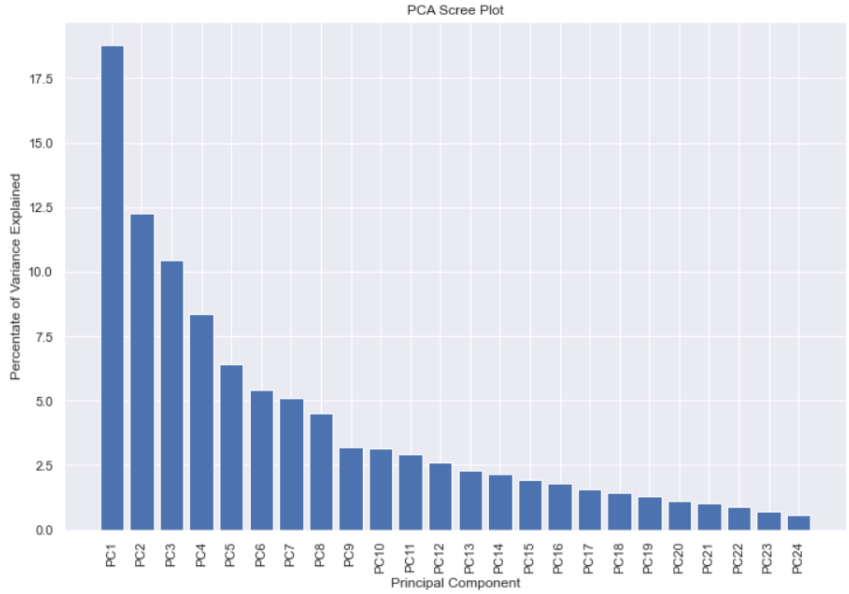
Dropping highly correlated features

Features dropped are 'exercise', 'friendly_strangers',

The reason for dropping these highly correlated features will skew the weights that we generated to predict new inputs.

Dimensionality Reduction using PCA

```
percent_variance = np.round(pca.explained_variance_ratio_* 100, decimals =2)
columns = ['PC1', 'PC2', 'PC3', 'PC4','PC5', 'PC6', 'PC7', 'PC8','PC9', 'PC10', 'PC11', 'PC12','PC13', 'PC14', 'PC15',
           'PC16','PC17', 'PC18', 'PC19', 'PC20','PC21', 'PC22', 'PC23', 'PC24']
plt.figure(figsize=(12, 8))
plt.bar(x= range(1,25), height=percent_variance, tick_label=columns)
plt.ylabel('Percentate of Variance Explained')
plt.xlabel('Principal Component')
plt.xticks(rotation=90)
plt.title('PCA Scree Plot')
plt.show()
print(percent_variance)
```

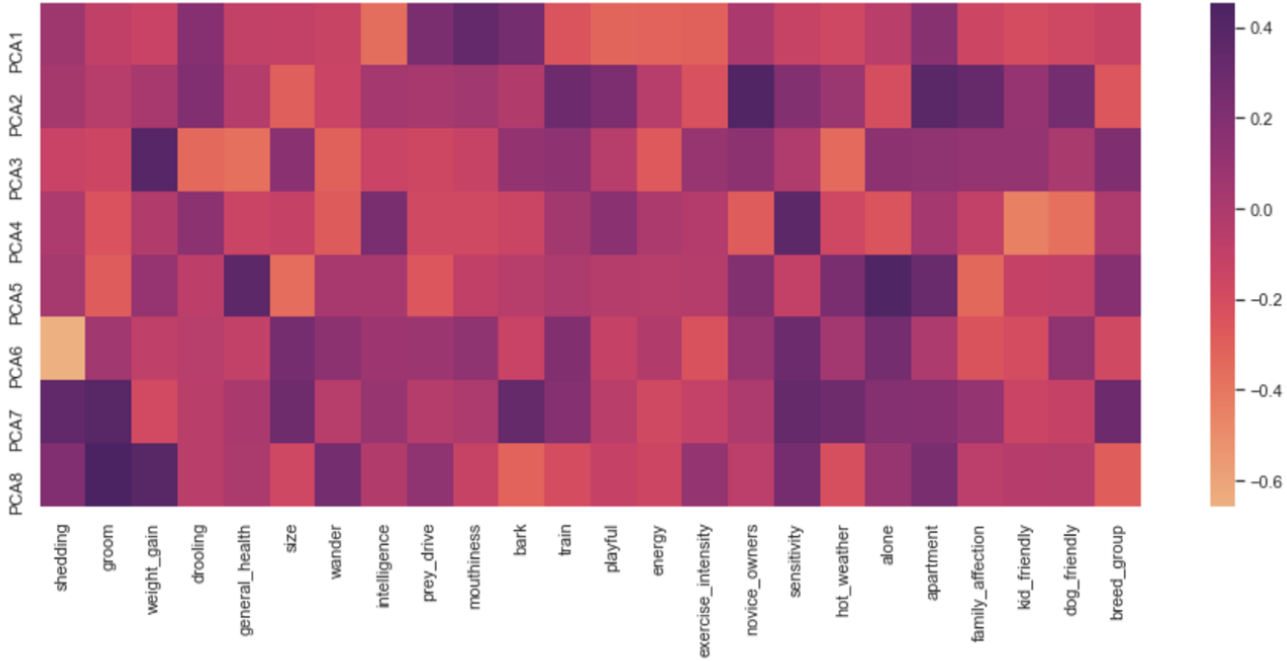



```
[18.77 12.27 10.45  8.36  6.41  5.43  5.12  4.53  3.2   3.15  2.94  2.62
  2.28  2.15  1.93  1.8   1.58  1.41  1.28  1.13  1.03  0.88  0.71  0.58]
```

8 eigenvector describes 70 percent of the variance

Dimensionality Reduction using PCA

```
plt.figure(figsize=(16,6))
ax = sns.heatmap(pca.components_[0:8],
                 cmap="flare",
                 yticklabels=[ "PCA"+str(x) for x in range(1,9)],
                 xticklabels=list(dataset.columns))
```



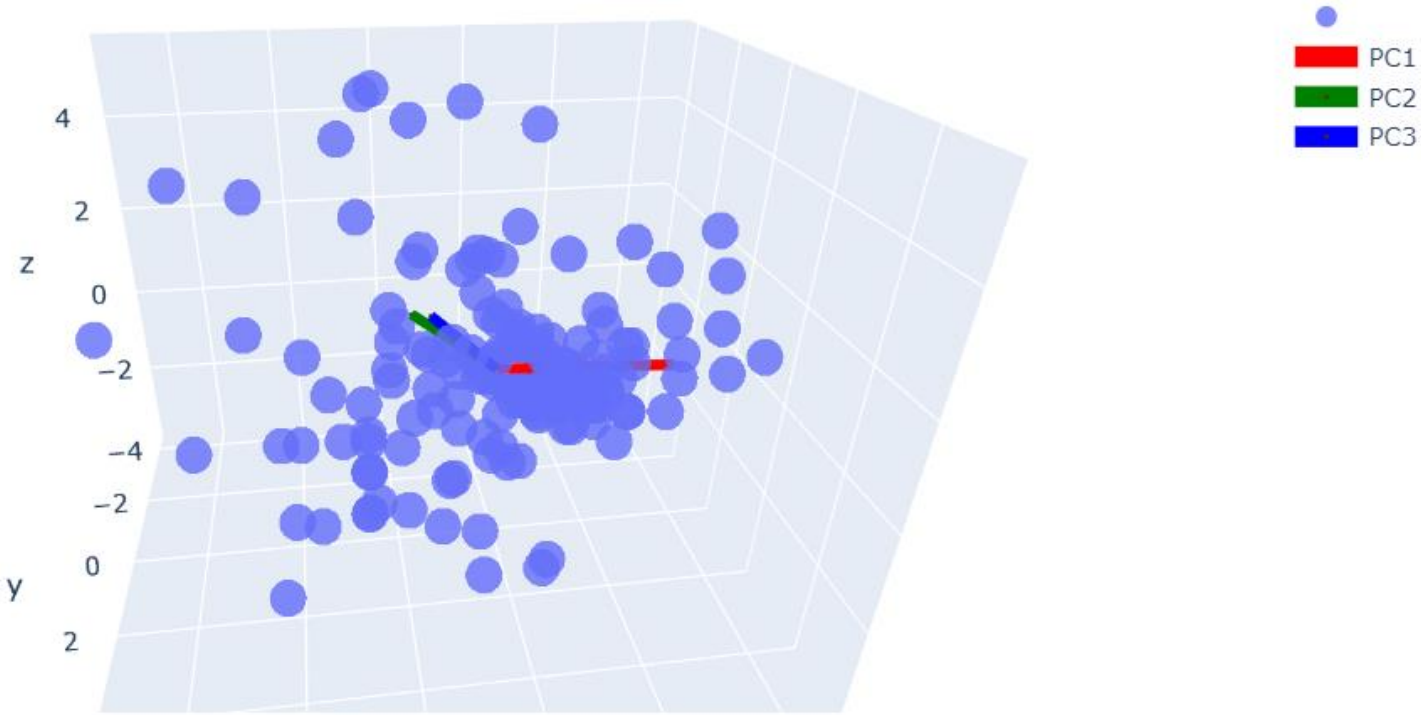
the weight of each feature in each eigenvector

Dimensionality Reduction using PCA

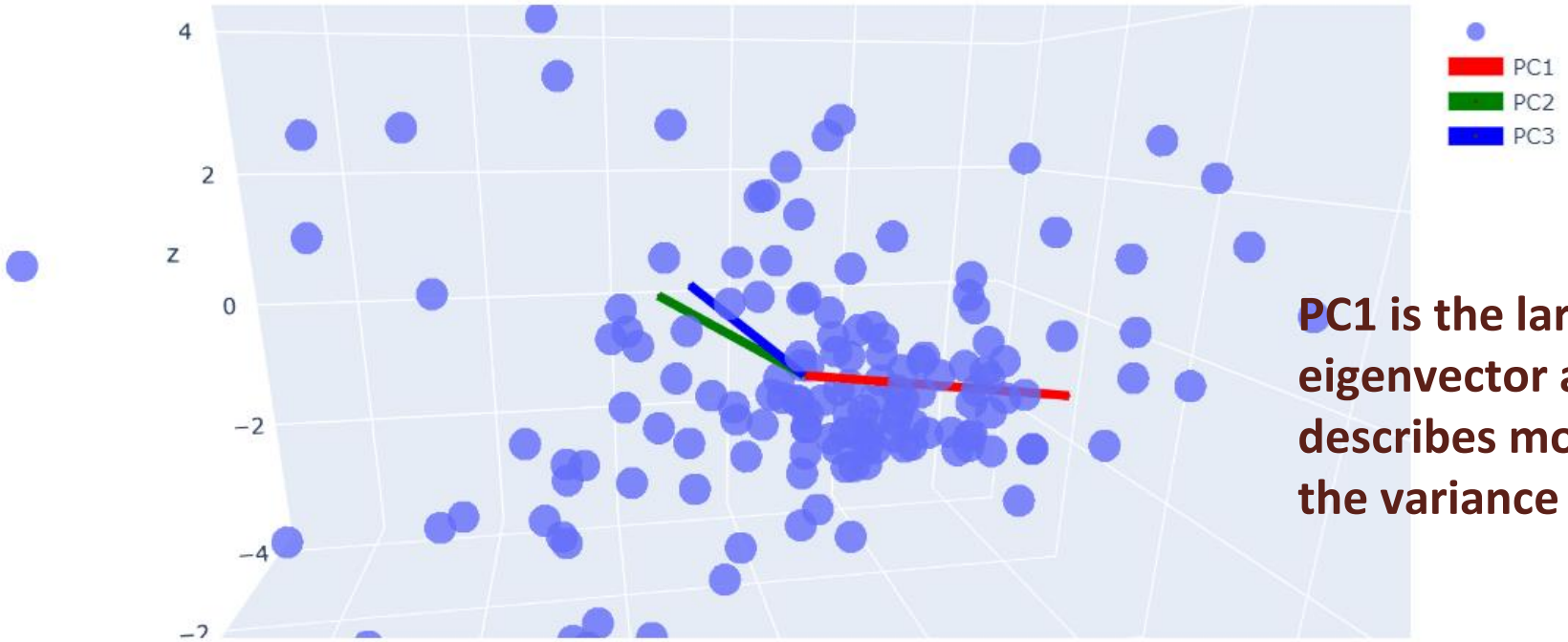
```
components = pd.DataFrame(pca.components_, columns=x.columns)
components.rename(index= lambda x:'PC-' + str(x+1), inplace=True)

# Top 3 positive contributors
pd.DataFrame(components.columns.values[np.argsort(-components.values,axis=1)[:,:3]],
             index=components.index, columns=['1st Max', '2nd Max','3rd Max'])
```

	1st Max	2nd Max	3rd Max
PC-1	mouthiness	bark	prey_drive
PC-2	novice_owners	apartment	family_affection
PC-3	weight_gain	breed_group	size
PC-4	sensitivity	intelligence	playful
PC-5	alone	general_health	apartment
PC-6	sensitivity	alone	size
PC-7	groom	shedding	bark

Dimensionality Reduction using PCA

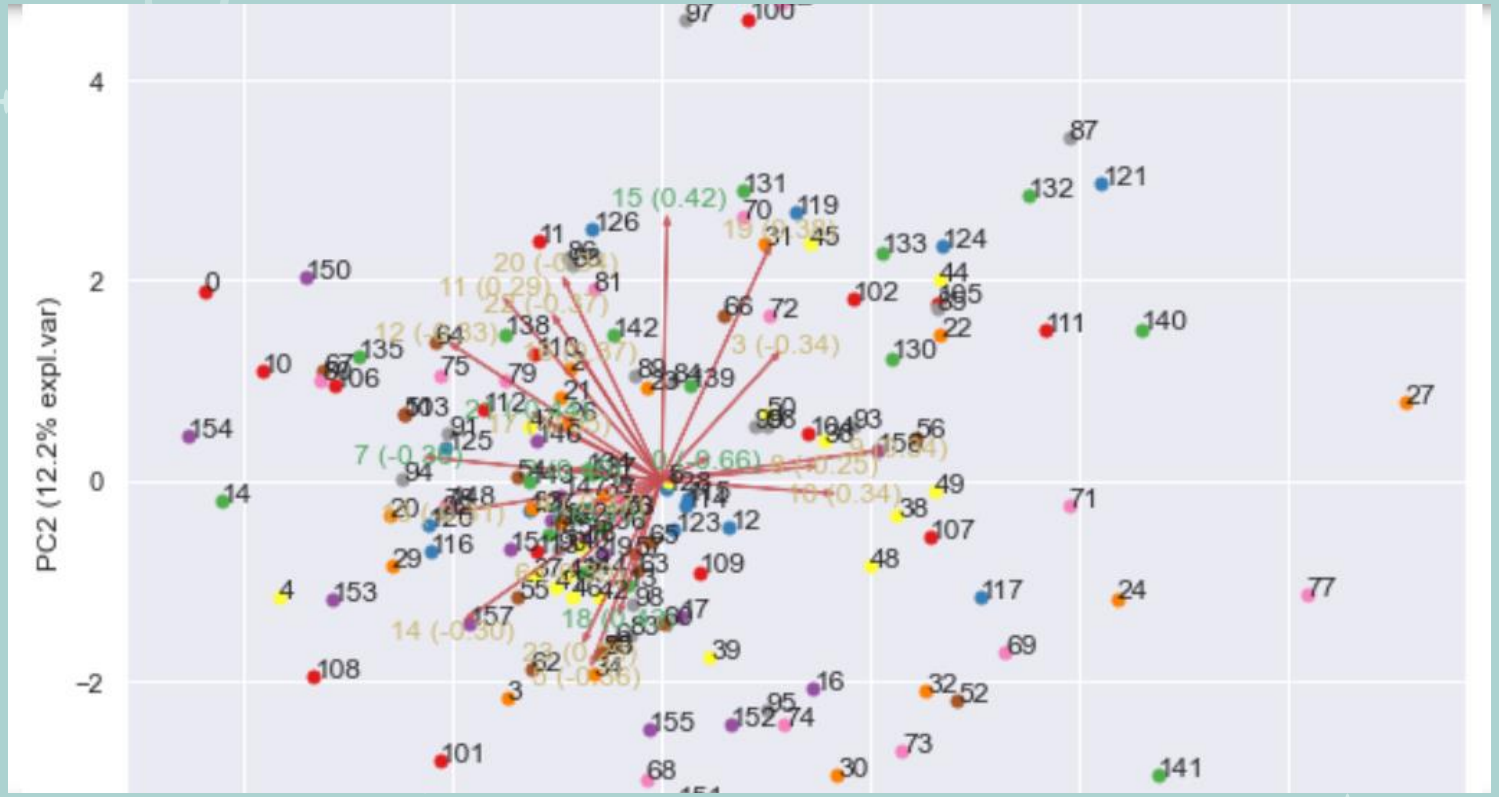


Dimensionality Reduction using PCA



PC1 is the largest eigenvector and describes most of the variance

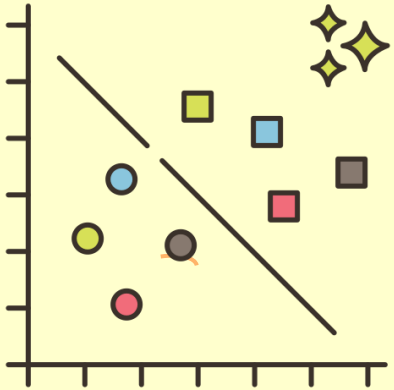
Dimensionality Reduction using PCA



A biplot gives a comparison of the relationship between all the features

03

Data Mining



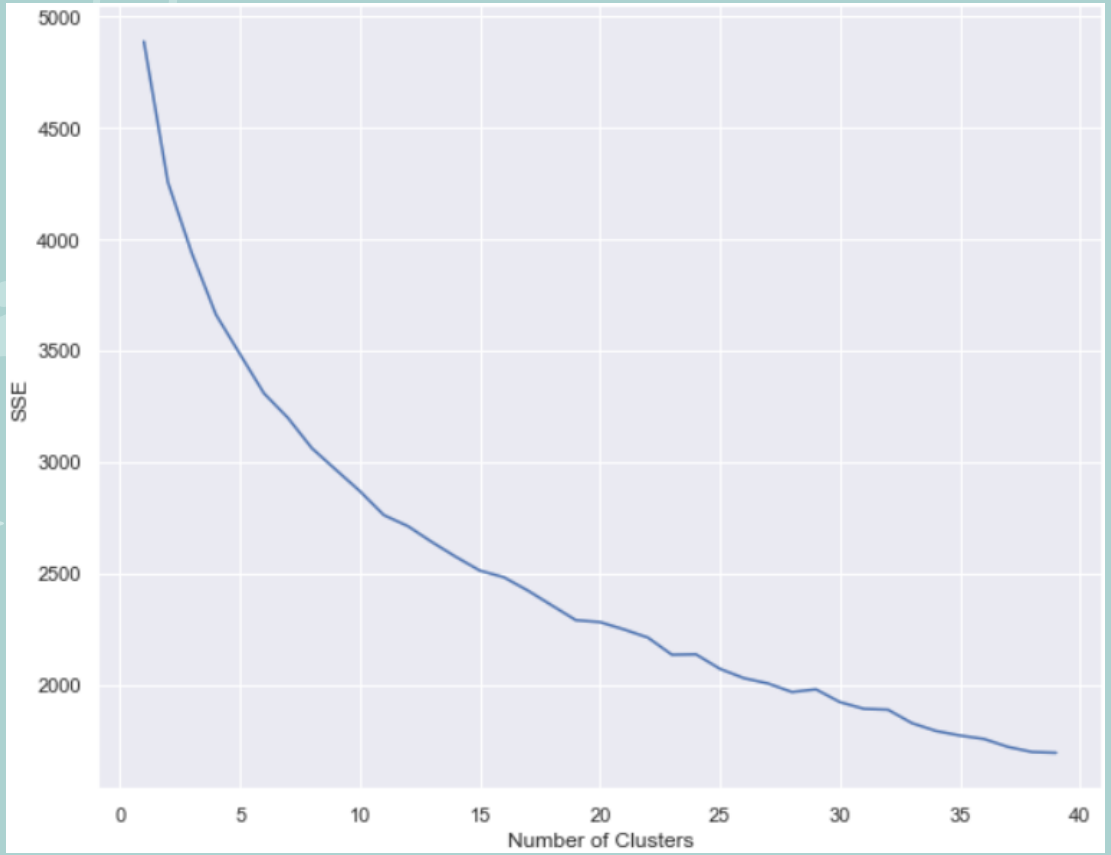
Clustering Vs Classification

Clustering finds out the similarity between different data and group them together. However, classification uses predefined classes or labels.

Classification will fail if new dog data comes in as every dog is unique.

Hence clustering is used to group similarity between different breeds.

K-means Clustering



The knee point is a point along the curve where the Sum of Squared Errors start decreasing linearly. We then use.

This point is used to determine an optimal value for the number of clusters in K-means clustering algorithm.

Knee Point, k = 13 clusters

This will group each data sample into one of the 13 clusters and give them a label in terms of Cluster ID.

Appending Cluster ID to the Dataset

	shedding	groom	weight_gain	drooling	general_health	size	wander	prey_drive	mouthiness	bark	...	family_affection	kid_friendly	dog_friendly	Cluster ID
Xoloitzcuintli	5	5	3	5	5	3	5	5	3	5	...	5	3	2	0
Italian Greyhound	4	5	1	5	2	1	4	5	4	3	...	5	1	3	0
Toy Fox Terrier	4	5	2	4	2	1	4	5	2	2	...	4	2	2	0
Saluki	4	4	1	5	4	4	5	5	4	4	...	5	3	2	0
Whippet	4	5	1	5	3	3	4	5	3	3	...	5	4	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Golden Retriever	2	4	4	3	2	4	2	3	1	2	...	5	5	5	12
Labradoodle	4	4	3	3	4	4	3	2	1	2	...	5	3	5	12
Brussels Griffon	2	3	3	3	3	1	5	1	3	2	...	5	5	3	12
Goldador	1	4	3	3	4	4	4	1	1	1	...	5	4	5	12
Boston Terrier	1	5	3	3	5	2	5	2	2	3	...	4	5	3	12

198 rows × 22 columns

Append the cluster labels to the dataset and sort them out according to clusters

How are they grouped?

C5

	shedding	groom	weight_gain	drooling	general_health	size	wander	intelligence	prey_drive	mouthiness	...	energy	exercise_intensity
Pekingese	1	1	3	5	2	1	1	2	5	4	...	2	1
Lhasa Apso	3	1	3	4	2	1	1	2	4	3	...	2	2
Shiba Inu	3	3	1	5	4	2	4	2	3	5	...	3	1
Dogue de Bordeaux	4	1	4	1	1	1	1	3	4	4	...	2	2
Peekapoo	2	2	4	5	3	2	1	3	4	4	...	3	2
Chinese Shar-Pei	5	1	4	5	1	3	2	2	2	5	...	2	1
Chow Chow	1	1	4	4	2	4	2	2	2	5	...	2	2
Maltese Shih Tzu	1	2	3	5	2	1	1	3	5	4	...	3	2
Japanese Chin	2	3	3	3	1	1	1	2	5	4	...	2	1

C6

	shedding	groom	weight_gain	drooling	general_health	size	wander	intelligence	prey_drive	mouthiness	...	energy	exercise_intensity
Rhodesian Ridgeback	3	5	3	4	2	4	4	3	4	3	...	3	3
Afghan Hound	2	1	1	5	3	4	5	3	5	3	...	3	3
Whippet	4	5	1	5	3	3	4	4	5	3	...	4	3
Xoloitzcuintli	5	5	3	5	5	3	5	5	5	3	...	3	3
Ibizan Hound	5	5	1	5	4	3	4	3	5	3	...	3	3
Saluki	4	4	1	5	4	4	5	3	5	4	...	4	2
Chinook	2	3	3	5	3	4	1	4	4	4	...	2	4
Azawakh	3	5	3	5	3	3	2	3	4	4	...	3	3
Borzoi	2	3	1	5	3	5	4	3	5	4	...	3	2
Basenji	4	4	2	5	3	2	5	3	4	2	...	3	1
Canaan Dog	3	3	2	5	4	3	2	4	3	3	...	4	3
Greyhound	3	5	2	5	3	4	4	3	5	4	...	3	2

The clusters are grouped based on similarity between each of the breed information.

04

Implementation

Dog Recommender System

```
Hi, I will be recommending 5 dog breeds to you today.
For each question, enter a number from 1 to 5.
1 = Strongly disagree. 5 = Strongly agree.

1. I do not mind a dog that sheds its fur often:
```

```
Hi, I will be recommending 5 dog breeds to you today.
For each question, enter a number from 1 to 5.
1 = Strongly disagree. 5 = Strongly agree.
1. I do not mind a dog that sheds its fur often: 2
2. I prefer a dog that is easy to groom: 5
3. I do not mind a dog that can gain weight easily: 3
4. I do not mind my dog drooling: 4
5. A dog that is resistant to illnesses is important: 2
6. I prefer a big dog: 4
7. I prefer an intelligent dog bred for jobs that require decision: 4
8. I prefer a dog that wanders around on it's own: 3
9. I prefer a dog that chases and hunts for small prey: 3
10. I do not mind a dog that likes chewing on things: 3
11. I prefer a dog that barks: 3
12. I want a dog that is easy to train: 4
13. I prefer a playful dog: 5
14. I prefer a dog with high energy and stamina: 3
15. I like taking my dog out for walks: 3
16. I have little to no experience raising dogs: 3
17. My home is generally quiet without loud sounds or distractions: 5
18. I would like to bring my dog out and excercise on a hot sunny day: 4
19. I prefer a dog that is able to be by itself and not crave attention: 3
20. I prefer my dog to be calm indoors and polite with strangers: 3
21. I prefer a dog that is affectionate with my family members: 5
22. The dog has to be friendly with small kids & children: 4
23. I would like my dog to be friendly and not dominate other dogs: 4
Inputs: [2, 5, 3, 4, 2, 4, 4, 3, 3, 3, 3, 4, 5, 3, 3, 3, 5, 4, 3, 3, 5, 4, 4]
According to your desired dog features, Top 5 Breeds to consider are as follows:
      Tibetan Mastiff
       Border Collie
   Shetland Sheepdog
    Belgian Malinois
 German Shepherd Dog
```

How it Works

Input

User inputs the 1-5 ranking of the desired dog features (5 - strongly agree)

Predict the Cluster

PCA and K-means clustering

Compares

Perform cosine similarity to find the similarities between user input and breeds

Ranks

Rank the similarity vectors in ascending order

Recommend

Display the top 5 related breeds

Demonstration of the System

05

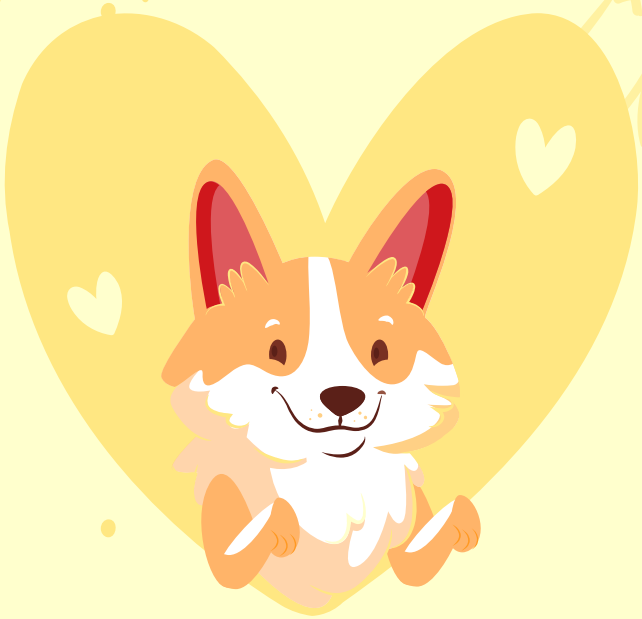
Conclusion

Summary

- Data Pre processing
  -feature selection
  -visualization
- Data Mining
  - Clustering
  -visualization
- Similarity metric
  - cosine similarity

Future Work

- Implement intuitive user interface
- Option for User-based collaborative filtering
- More available dog breeds recognized by the FCI (World Canine Organization)



THANK YOU

Do you have any questions?