AnalyticsGrads Hackathon 2024

[Link to Github Repository]

Problem Statement:

Behavior Classification: Develop a model to classify a subset of annotated behaviors from accelerometer data. This task involves creating a sophisticated algorithm to accurately identify different behaviors based on data patterns.

Introduction

The main reason why we, as a group, chose this problem statement over the other is because we all have experience in coding and most of us have experience in developing AI models in college at a basic level. This is why we preferred this over the non-coding path, we felt the area we work in already revolves around coding everyday. We wanted to challenge ourselves and combine all our skills and knowledge together to work on this fun problem. It also gives us an opportunity to work with each other and learn from each other, as we have varying levels of experience. This problem statement is also very interesting in analyzing specific behaviors in dogs.

For the problem statement we decided on behavior characterization. Because the dataset was so large we felt it would be more beneficial to focus on a subset of behaviors. We wanted to specifically focus on different movement types, and avoid classifying emotions. The data is split into movement types and sub movement types, such as behavior with ID 1 is resting and ID 1-2 is sitting down. So we did not focus on more than one sub movement within a movement. We decided on the following behavior types:

X1	Tightening up/Adjusting/Moving in same position
20-0	Drinking (key behavior)
1-2	Standing
2-0	Walking
23-2	Tail wagging medium

Our Approach

The approach we took to solve this problem was to build various classification algorithms, compare them, and to build a final model to complete this task. Before this though, we explored the data and began to clean it to prepare ir for our model

One of key findings that we found while exploring the data was the high occurrence of NULL values in the x, y and z columns. One potential solution was to replace these null values with 0's, but with some experimentation, we found it to be best practice to remove any rows with null values.

```
null_x = training_set_exploded["x"].isnull().sum()
null_y = training_set_exploded["y"].isnull().sum()
null_z = training_set_exploded["z"].isnull().sum()
null_labels = training_set_exploded["labels"].isnull().sum()

null_x, null_y, null_z, null_labels
```

```
(234032, 234032, 234032, 0)
```

We also exploded the "labels" column because some labels were arrays, meaning that there were 2 or more values in the labels for one cell. We knew this would be a problem in the future when developing and training our models so we created new rows for each value inside the labels array, using the pandas explode method. For example if there were 2 values in the labels array, 2 new rows would be created for each value, with the same x.y.z value but different label values. After exploding the labels, the data was now ready for a classification model.

Pre-Processing

[4]:
```
#some rows have 2 or more labels, use explode so each row has only 1 label
training_set_exploded = training_set.explode('labels')
testing_set_exploded = testing_set.explode('labels')
```

[5]:
```
training_set_exploded
```

[5]:

	timestamp	x	y	z	labels	filename
13497564	0.010	0.219971	-2.150879	-1.247314	SM	aadi_ga_20150123_1.parquet
26846227	0.020	0.124756	-1.658203	-0.735352	SM	aadi_ga_20150123_1.parquet
6305228	0.040	0.148926	-1.443359	-0.931641	SM	aadi_ga_20150123_1.parquet
30245674	0.060	0.139893	-1.896484	-1.113281	SM	aadi_ga_20150123_1.parquet
11300293	0.080	0.358154	-2.125977	-1.261963	SM	aadi_ga_20150123_1.parquet
...	...	...	...	...	...	...
159516	622.109	-0.359375	0.531250	-0.750000	SM	zwicky_ga_20150629_1.parquet
27452467	622.129	-0.843750	-0.234375	-1.218750	SM	zwicky_ga_20150629_1.parquet
644896	622.139	-1.156250	-1.546875	-1.406250	SM	zwicky_ga_20150629_1.parquet
11101991	622.159	0.546875	-6.406250	-0.921875	SM	zwicky_ga_20150629_1.parquet
10347225	622.169	0.140625	-8.000000	-0.875000	SM	zwicky_ga_20150629_1.parquet

79732876 rows × 6 columns

We knew this would significantly increase the number of data so after that we dropped duplicates using pandas to remove any duplicate values within the dataset.

```
training_set_exploded = training_set_exploded.drop_duplicates()

testing_set_exploded = testing_set_exploded.drop_duplicates()
```

```
training_set_exploded.shape
```

```
(79729451, 6)
```

When printing out the unique labels for both datasets, we identified that there were whitespaces entries in the ‘labels’ column. To enhance data processing efficiency and ensure consistency throughout our datasets, we will trim these whitespaces.

```
# Trim whitespace from 'labels' column in both datasets
training_set_exploded['labels'] = training_set_exploded['labels'].str.strip()
testing_set_exploded['labels'] = testing_set_exploded['labels'].str.strip()
```

Before trimming the whitespaces from the ‘labels’ column:

```
Name: labels, Length: 121, dtype: int64
Unique labels in training set: ['H' 'NULL' '2-4' '27-0' '20-0' '21-2' '1-2' '44-0' '33-0' '2-0' '40-6'
 '5-1' '1-1' '1-C' '35-0' '23-2' '46-0' '21-1A' '22-2' '3-2' '3-1' '21-5'
 '19-1' 'P' '5-5' '3-4' '30-0' '1-C2' '30-1' '1-C1' '48-0' '28-0' '35-1'
 '1-A2' '12-C2' '29-3' '23-3' '36-0' '1-B2' '1-3' '21-1D' '23-1' '1-A1'
 '43-0' '32-0' '1-U' '1-B1' '45-0' '12-B1' 'SM' ' 1-C1 ' '12-B2' '21-4'
 '40-2' 'X1' '21-1C' '4-1' '5-2' '5-3' '19-2' '50-0' '40-5' '31-0' '23-4'
 '21-1' 'S' '2-7' '2-6' '22-1' '29-1' '12-A1' 'H ' ' 1-C' ' 1-A1 ' '2-5'
 '29-4' '3-0' '5-1  ' '29-0' '40-4' '29-2' '41-0' '26-0' '1-1   ' '3-3'
 '1-C2  ' '1-2 ' '1-2       ' '12-A2' '00' '1-B1  ' '21-1B' '40-1' '29-6'
 '1-4' '26-2' '2-3B' '34-1' '37-0' '43-1' '5-4' '35-2' '2-6A' ' 2-0'
 '1-2  ' '3-5' '5-5  ' '21-0' '1-5' '38-0' '1-1 ' ' 1-C2 ' '20-0 ' '2-3A'
 '21-3' '23-0' '4-2' '39' '2-8' '42-0' '40-3' '1-2   ' '5-5 ' '26-1'
 '40-7' '2-1' '21-2D' '2-2' '2-3' 'Unknown label' '1-C  ' '2-4  ' '2-4 '
 ' 1-2 ' '2-0 ' ' 2-4' '5-1 ' ' 1-1 ' '4-0' ' 5-5' '28' '5-55' 'H  '
 '29-5' '3-6' '46' '  2-4' ' 5-1' ' 2-4 ' '34-0']
```

After trimming the whitespaces from the ‘labels’ column:

```
Name: labels, Length: 102, dtype: int64
Unique labels in training set: ['H' 'NULL' '2-4' '27-0' '20-0' '21-2' '1-2' '44-0' '33-0' '2-0' '40-6'
 '5-1' '1-1' '1-C' '35-0' '23-2' '46-0' '21-1A' '22-2' '3-2' '3-1' '21-5'
 '19-1' 'P' '5-5' '3-4' '30-0' '1-C2' '30-1' '1-C1' '48-0' '28-0' '35-1'
 '1-A2' '12-C2' '29-3' '23-3' '36-0' '1-B2' '1-3' '21-1D' '23-1' '1-A1'
 '43-0' '32-0' '1-U' '1-B1' '45-0' '12-B1' 'SM' '12-B2' '21-4' '40-2' 'X1'
 '21-1C' '4-1' '5-2' '5-3' '19-2' '50-0' '40-5' '31-0' '23-4' '21-1' 'S'
 '2-7' '2-6' '22-1' '29-1' '12-A1' '2-5' '29-4' '3-0' '29-0' '40-4' '29-2'
 '41-0' '26-0' '3-3' '12-A2' '00' '21-1B' '40-1' '29-6' '1-4' '26-2'
 '2-3B' '34-1' '37-0' '43-1' '5-4' '35-2' '2-6A' '3-5' '21-0' '1-5' '38-0'
 '2-3A' '21-3' '23-0' '4-2' '39' '2-8' '42-0' '40-3' '26-1' '40-7' '2-1'
 '21-2D' '2-2' '2-3' 'Unknown label' '4-0' '28' '5-55' '29-5' '3-6' '46'
 '34-0']
```

We also explored other approaches such as imputing but found that it would have been more complex than dropping nulls so we went for the dropping nulls instead and removing outliers using interquartile ranges but we found that these disimproved our model accuracy so decided against it.

Findings (Report of finding and explanation as to how the prototype solves the given problem):

A Decision Tree classifier makes decisions based on feature values, splitting the data at each node according to specific criteria to reach a decision at the leaf nodes. We used grid search to find the best hyperparameters.

After experimenting with hyperparameters, we applied a bagging technique to counter potential variance in the model. After this, we achieved a weighted F1 Score of 69.02% with the following parameters, the best accuracy we got out of all the models.

We believe our experiments show that there is a definite possibility for this type of sensor data to be equipped to examine dogs movement. While our accuracy was not extremely high, it was more accurate than pure guesswork. This work could be expanded by introducing more types of movement into our models and with more compute power we could train a more complex model to make classifications like what we have done. Another potential improvement would be if we had even more data available to us for training our model.

This was the configuration of our

```
= {'criterion': 'gini', 'max_depth': 15, 'min_samples_leaf': 2, 'min_samples_split': 20, 'random_state':42}
```