

# Data for Good: Connecting Graduate Coursework with Real-World Solutions for Nonprofits

**SecondHand Hounds**

*Vasudha Gulati*

*Shalini Sarathy Komandur Chakravarthy*

*Andrei Jula*

**2023 Great Lakes Data & Analytics Summit**



# Shalini Sarathy Komandur Chakravarthy

Data Flow / Preprocessing

PowerBI - Application Charts / KPIs / Metrics / Insights

# SHH Organizational Sponsors & Oakland University Team

- **Carrie Openshaw**, Program Director – “Oversees Program and Outreach Departments as well as shelter partner relationships”
- **Maggie Schmitz**, Director of Marketing – “Oversees Marketing Department, manages SHH social media, manages media / PR inquiries and marketing requests”
- **Tashina Schumacher**, Chief of Staff – “Oversees Operations, Human Resources & Finance”



Left to Right:

Shalini Sarathy Komandur Chakravarthy  
James Balwinski  
Dr. Vijayan Sugumaran  
Phat Pham  
Zahra Khan  
Andrei Jula  
Vasudha Gulati

# Animal Rescue as a Nonprofit Organization

- The **SecondHand Hound** – a nonprofit animal rescue organization, founded in 2009
- The organization mainly focuses on finding a home for pets that can no longer be provided care for by their owners, or rescues from kill shelters, in the Midwest area
- During the process, the volunteers provide a temporary safe shelter, veterinary care, rehabilitation or hospice care
- All the involved project activities related to adoption and neonatal programs (pets born with health issues or disabilities), are coordinated by a dedicated and inspired staff
- Some of the activities are collecting donations, creating awareness, identifying and reaching out to possible foster to adopting clients, organizing fostering and volunteering events.

# How it all started..

- Bringing their data together
- Simple intake and outcome excel report.
- Find a permanent home for rescued animals as soon as possible.

# Requirement and Outcome

LIVE INTAKE JAN				
CANINES	Adult	<5mo	unkown	Subtotal
Stray	1			1
OS	128	39		167
OSeuth				0
Transfer Instate				0
Transfer Outstate				0
Transfer International				0
Seized				0
Other		34		34
FELINES	Adult	<5mo	unkown	Subtotal
Stray	2			2
OS	33	16		49
OSeuth				0
Transfer Instate	5	11		16
Transfer Outstate				0
Transfer International				0
Seized				0
Other				0
LIVE OUTCOMES JAN				
CANINES	Adult	<5mo	unkown	Subtotal
Adoption	74	50		124
Returned to Owner				0
Transferred in State				0
Transferred Out of State				0
Transferred International				0
Returned to Field				0
Other				0
FELINES	Adult	<5mo	unkown	Subtotal
Adoption	14	14		28
Returned to Owner				0
Transferred in State				0
Transferred Out of State				0
Transferred International				0
Returned to Field				0
Other				0

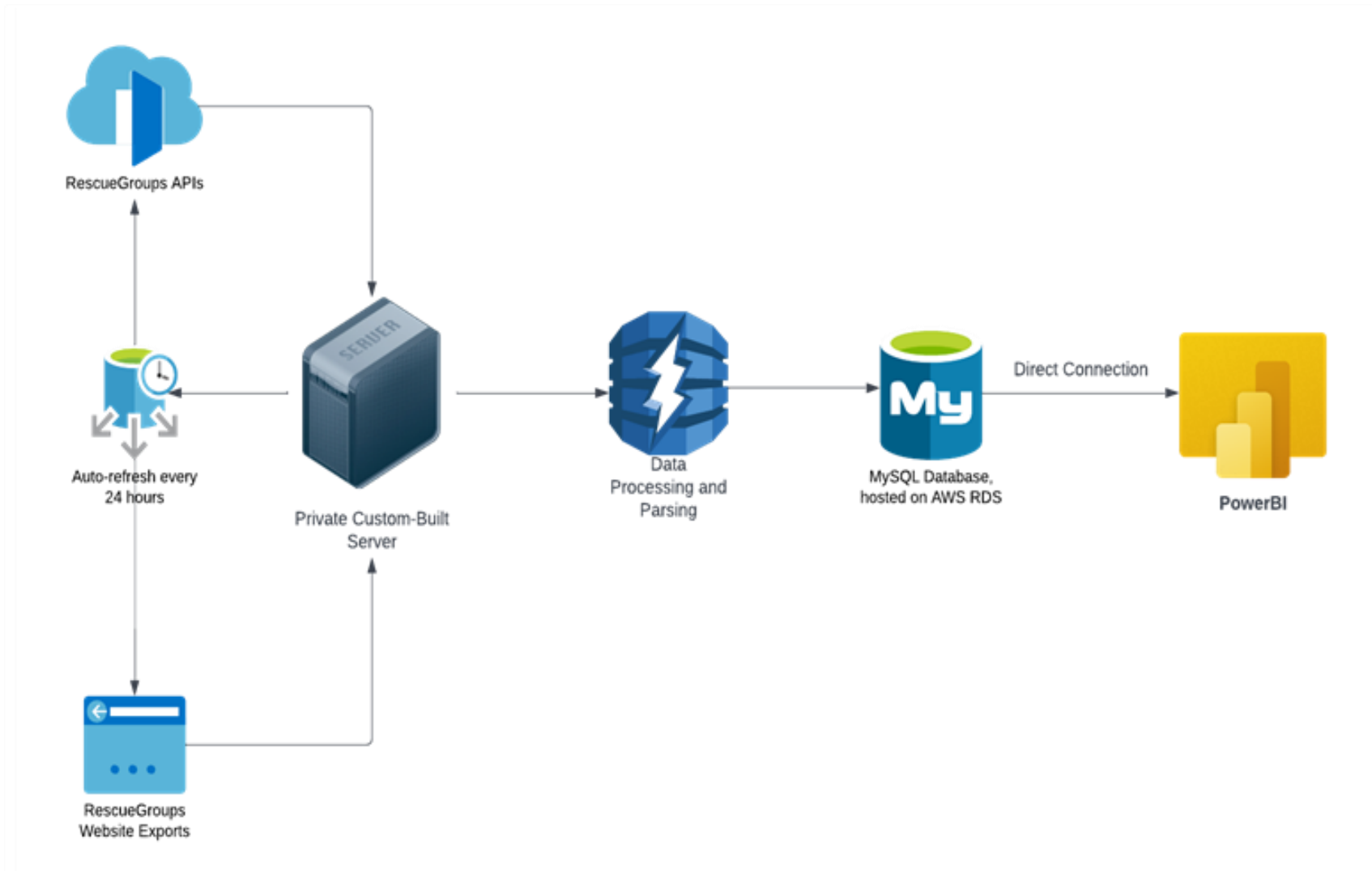
Year							Month										
Select all	2017	2018	2019	2020	2021	2022	Select all	1	2	3	4	5	6	7	8	9	10
Year	Month	Species	Stray at Large Adult	Stray at Large upto 5 Months	Stray at Large Age Unknown	Relinquished by Owner Adult	Relinquished by Owner Up To 5 Months	Relinquished by Owner									
2022	1	Canine			1												
2022	1	Feline	1														
2022	2	Canine															
2022	2	Feline	1														
2022	3	Canine															
2022	3	Feline															
2022	4	Canine															
2021	1	Canine															
2021	1	Feline	1														
2021	2	Canine															
2021	2	Feline															
2021	3	Canine															
2021	3	Feline															
2021	4	Canine			1												
2021	4	Feline	1														
2021	5	Canine															
<b>Total</b>			<b>79</b>	<b>352</b>													

# Initial Challenges

1. Data Inconsistencies - Many nulls, headers named differently per different sources for joining, merged information (Ex: multiple pet names in single cell), and many more.
2. Historical Data - Rescue Groups does not hold information prior to updates. (Ex: Who was the last foster if different?)
3. The ETL Process: Gaining access to create an API and converting files
4. Connecting the data to PowerBI (no auto-refresh)
5. Local vs Client (Ex: Python code may work locally but not for client)



# ETL Solution Implementation





## Portions of the server's code

```
src > schedulers > scheduler_secondHandHounds.js > ...
019 // Region Function - Import Foster/Adopter ID From XML
020 function importFosterAdopterFromXML()
021 {
022   return new Promise((resolve) => {
023     // Log time
024     const startedAt = moment().unix();
025
026     // Fetch data
027     apiWrapperXML({ type: 'animals', viewID: '558558', tagName: 'Animal_' })
028     .then((json) => {
029       // See how attribute names with DB attribute names
030       let attributeMap = {
031         'id': 'id',
032         'Adopter_ContactID': 'adapterID',
033         'Foster_ContactID': 'fosterID'
034       };
035
036       // Go over all items in the JSON array and parse them to the correct format we need
037       let sqlValues = [],
038           keyList = Object.keys(attributeMap),
039           dbAttributeList = keyList.map(key => attributeMap[key]);
040
041       json.forEach(item => sqlValues.push(`${keyList.map(attributeKey => parseAttributeValue(item[attributeKey]
042
043 // Generate SQL statement
044 let sqlStatement =
045   `INSERT INTO Animals (${dbAttributeList.join(',')} )\n` +
046   `VALUES\n` +
047   `sqlValues.join(',\n') +
048   `;\n\n DUPLICATE KEY UPDATE\n` +
049   `dbAttributeList.map(item => (item + ` = VALUES(` + item + `))\n).join(',\n');
050
051 // Invoke query statement to DB
052 let invokeDBWrite = function()
053 {
054   const connection = getMySQLConnection();
055   connection.query(sqlStatement, (error) => {
056     // Notify via Slack
057     if (error) sendSlackMessage('SQL Error Import Foster/Adopter From XML', error.sqlMessage);
058     else sendSlackMessage('Done', '[Job Completed] Import Foster/Adopter From XML, took ${moment().unix() -
059
060 // Close connection
061 connection.end();
062
063 // Resolve
064 resolve('Done');
065
066 });
067
068 // Write to file for review
069 if (isLocal())
070 {
071   writeFile('./tempData/secondHandHounds_FosterAdopter.sql', sqlStatement)
072   .then(() => invokeDBWrite())
073   .catch(writeError => {
074     invokeDBWrite();
075     sendSlackMessage('Write File Error Import FosterAdopterFromXML', writeError.message);
076   });
077 }
078 }
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

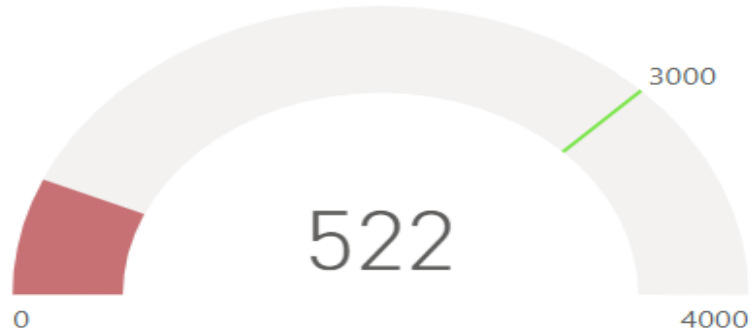
# Vasudha Gulati

PowerBI - Application Charts / KPIs / Metrics / Insights

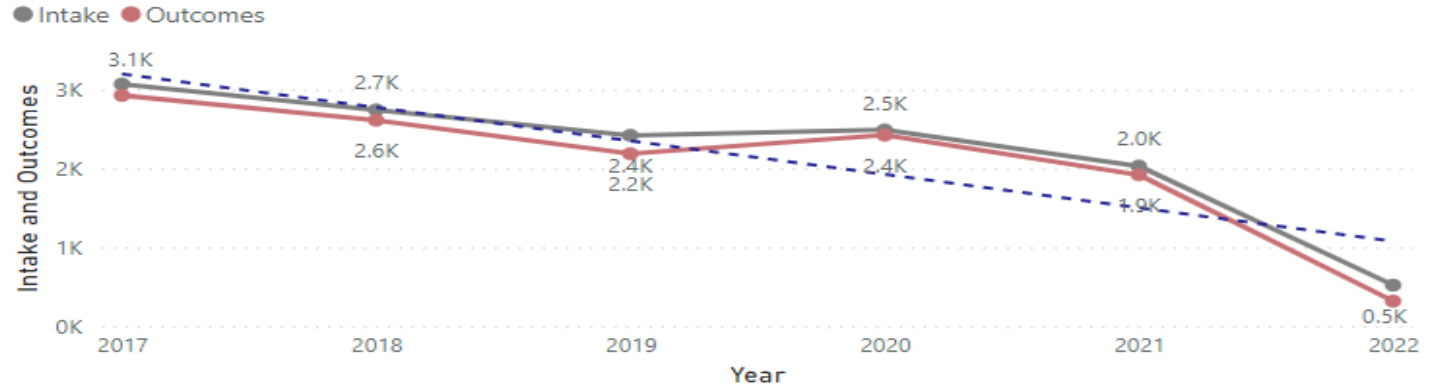
# Performance Dashboard



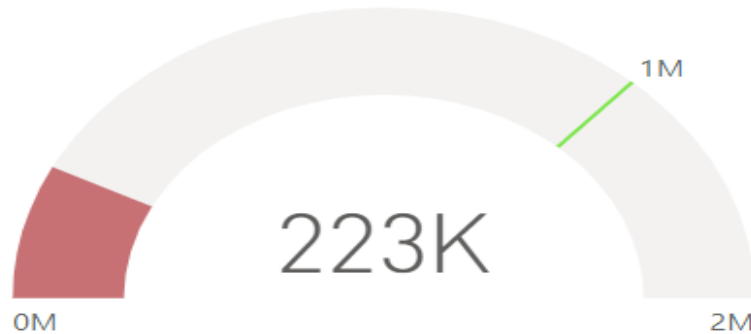
Animals Rescued/Intake 2022



Animal Intake Vs Outcomes Snapshot



Adoption Fees 2022



Animal Adoption Snapshot by Age

**127**  
Adult Dogs Adopted YTD

**14**  
Adult Cats Adopted YTD

**166**  
Puppies Adopted YTD

**14**  
Kittens Adopted YTD

**17**  
Avg Intake Age (m)

**20**  
Avg Outcome Age (m)

Year    
 Multiple selections

Quarter    
 All

Month    
 All

Species    
 Select all Cat Dog

Animal Intake   
 **13.27K**

Animals Adopted   
 **12.39K**

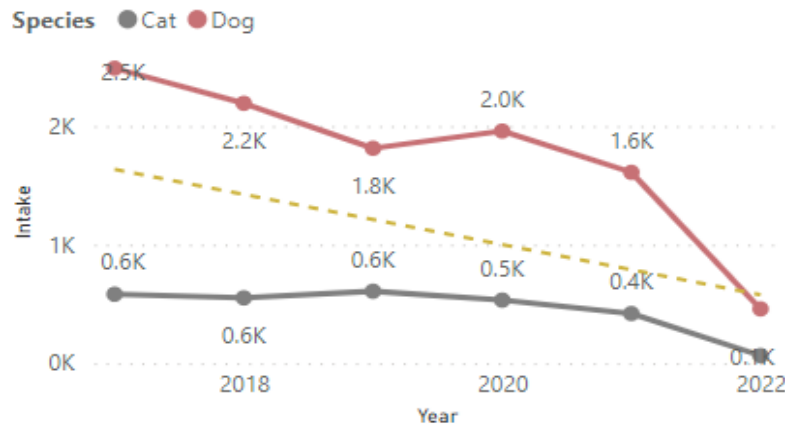
Adoption Fees   
 **5M**

Last Intake Date   
 Saturday, April 09, 2022

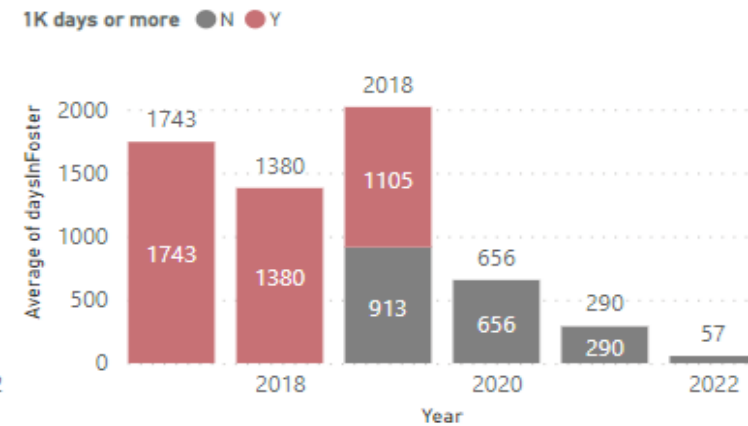
**Live Intakes** (by Category and Age Group)

Category	Adult	Baby (Less < 5 months)	Total
Transfer Outstate	1121	1142	2263
Transfer International	121	72	193
Transfer Instate	316	368	684
Stray	82	349	431
Other	2325	1548	3873
OS	2659	2631	5290
<b>Total</b>	<b>6739</b>	<b>6535</b>	<b>13274</b>

**Animals Adopted Yearly Comparison**



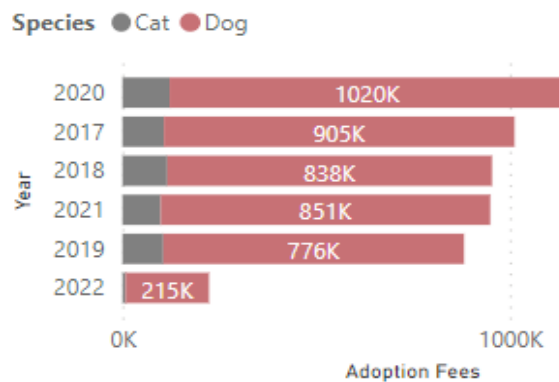
**Days in Foster**



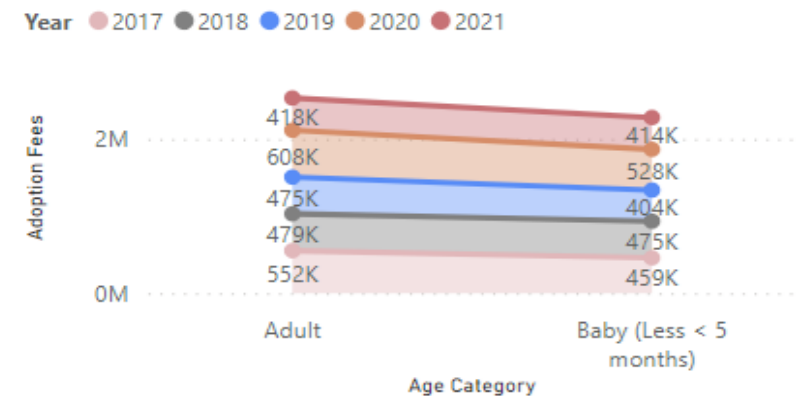
**Live Outcomes** (by Category and Age Group)

Category	Adult	Baby (Less < 5 months)	Total
Transfer Outstate	1114	1056	2170
Transfer International	121	61	182
Transfer Instate	316	337	653
Stray	81	320	401
Other	2298	1354	3652
OS	2642	2437	5079
<b>Total</b>	<b>6658</b>	<b>5734</b>	<b>12392</b>

**Adoption Fee Intake YOY**



**Adoption Fee breakdown by Age Category**



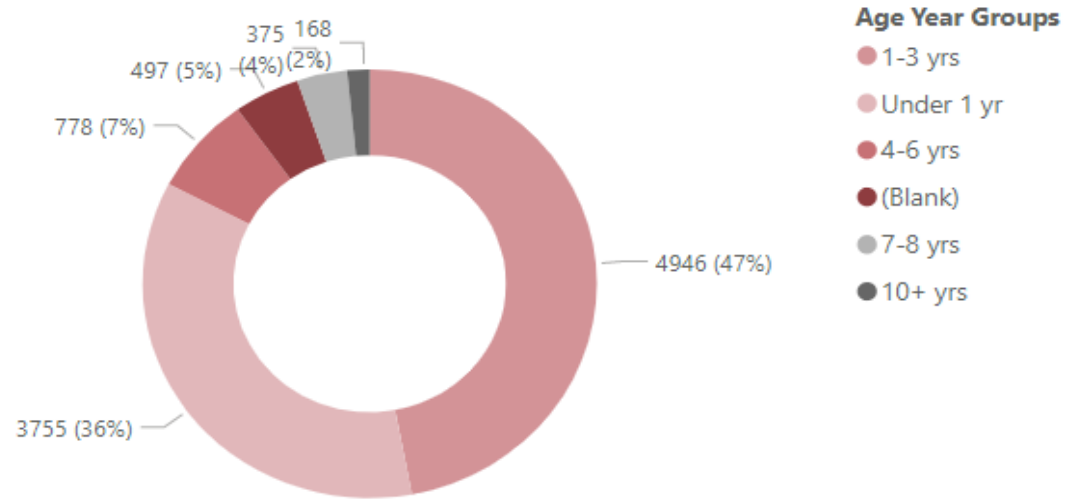
Year  
Multiple selections

Quarter  
All

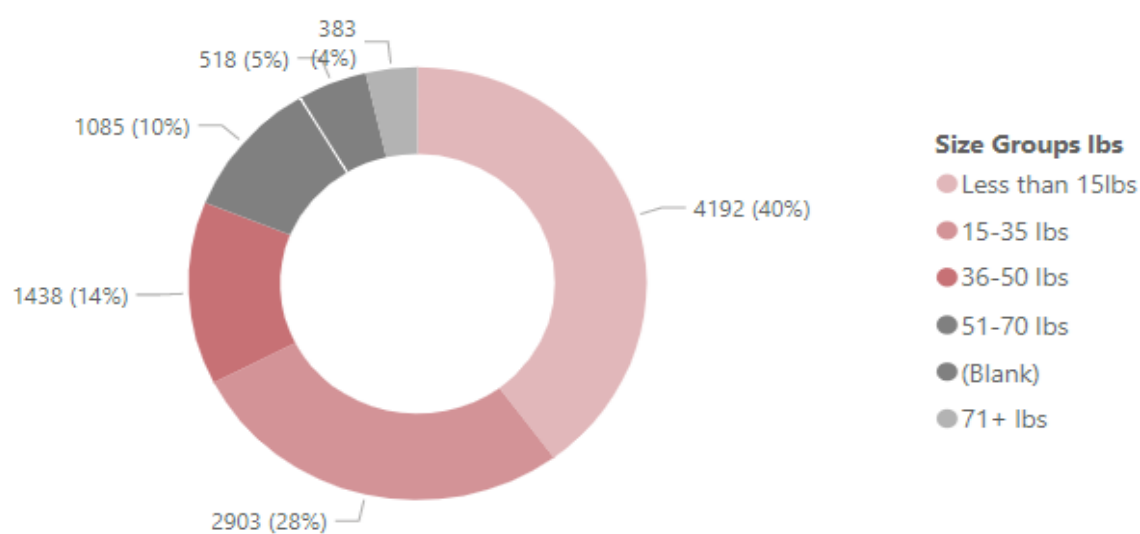
Month  
All

Species  
Select all Cat Dog

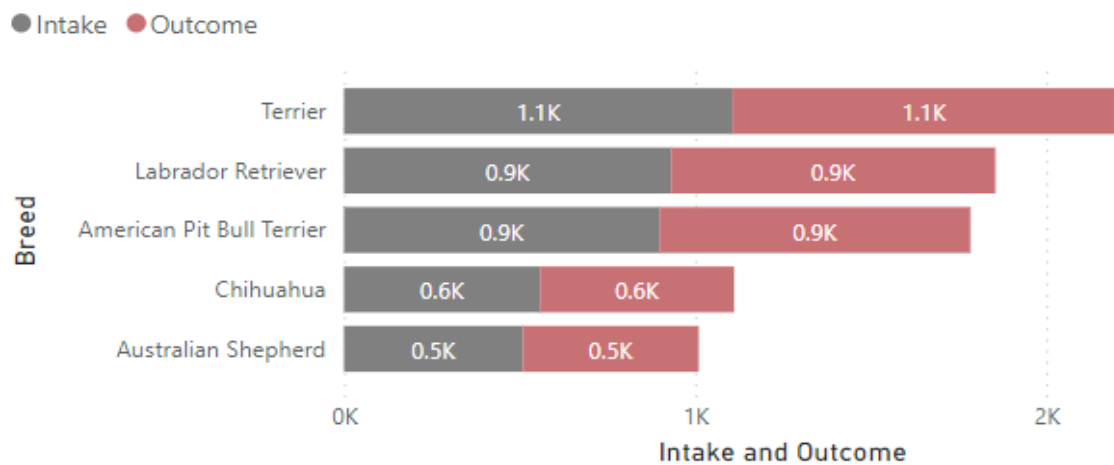
Animal Intake breakdown by Age Groups



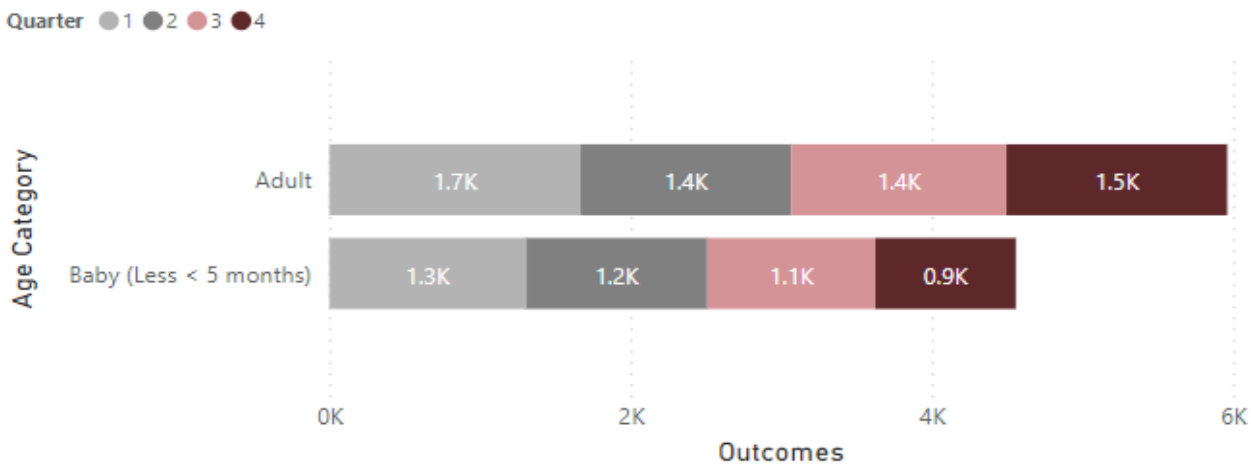
Animal Intake Breakdown by Size Groups



Animal Popularity by Breeds



Animal Outcome Trends by Age Category



# Current Dashboard

Receive Date:  2022,  2021,  2020,  2019,  2018  
 Status:  Adopted,  Available,  Euthanized  
 Size:  Large,  Medium,  Small,  X-Large  
 Age Group:  Adult,  Baby,  Senior  
 Species:  Bird,  Cat,  Chinchilla,  Dog,  Ferret  
 Breed:  Abyssinian,  Affenpinscher,  Afghan Hound,  African,  African Grey,  Airedale Terrier  
 Animal ID:  (Blank),  C110001,  C110002,  C110003,  C110004,  C110005

22,163

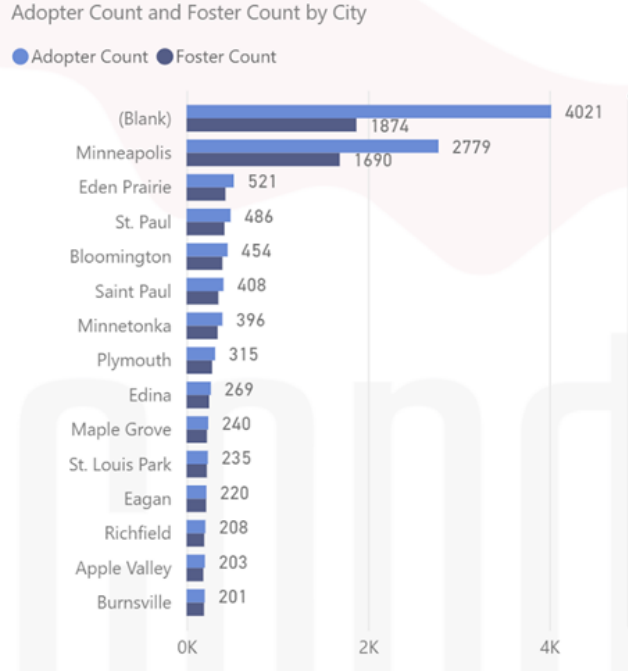
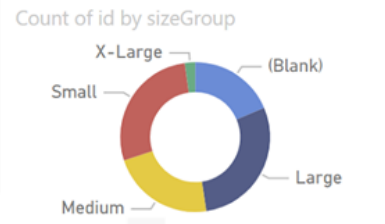
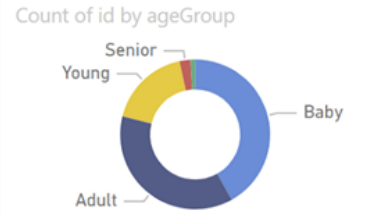
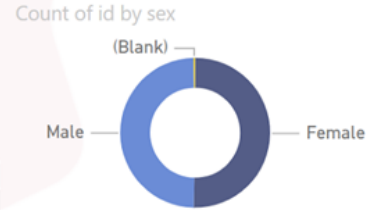
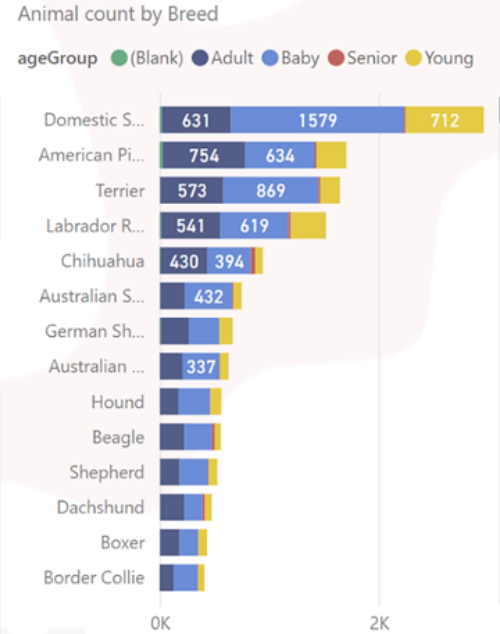
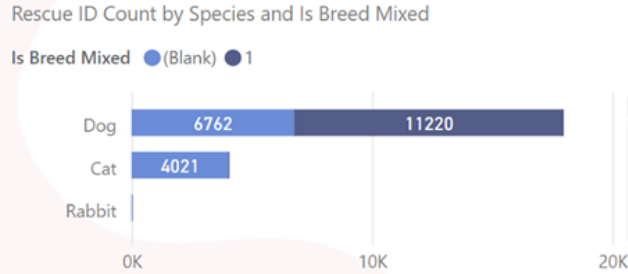
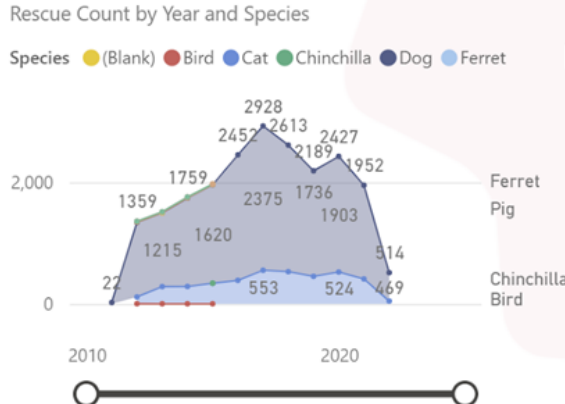
SHH Rescues

2,659

eVet Clients

54,365

Raiser's Edge Clients



Received Date	Status	Species	Breed	Name & ID	Birthdate	Age	Current Size
03/04/2022	Hold	Dog	Beagle	GingerSnap D220429	04/04/2020	928	18
03/04/2022	Hold	Dog	Beagle	GingerSnap D220429	04/04/2020	928	18
03/04/2022	Adopted	Dog	Australian Shepherd	Peebles D220430	27/01/2022	265	9
<b>Total</b>							<b>2505</b>



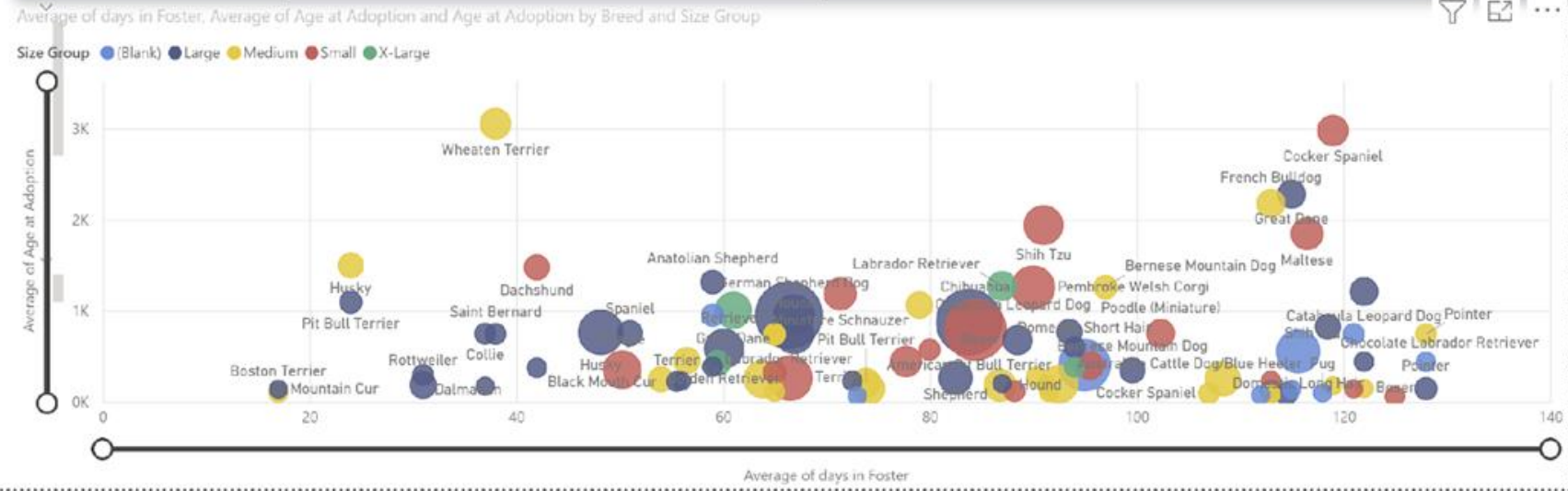
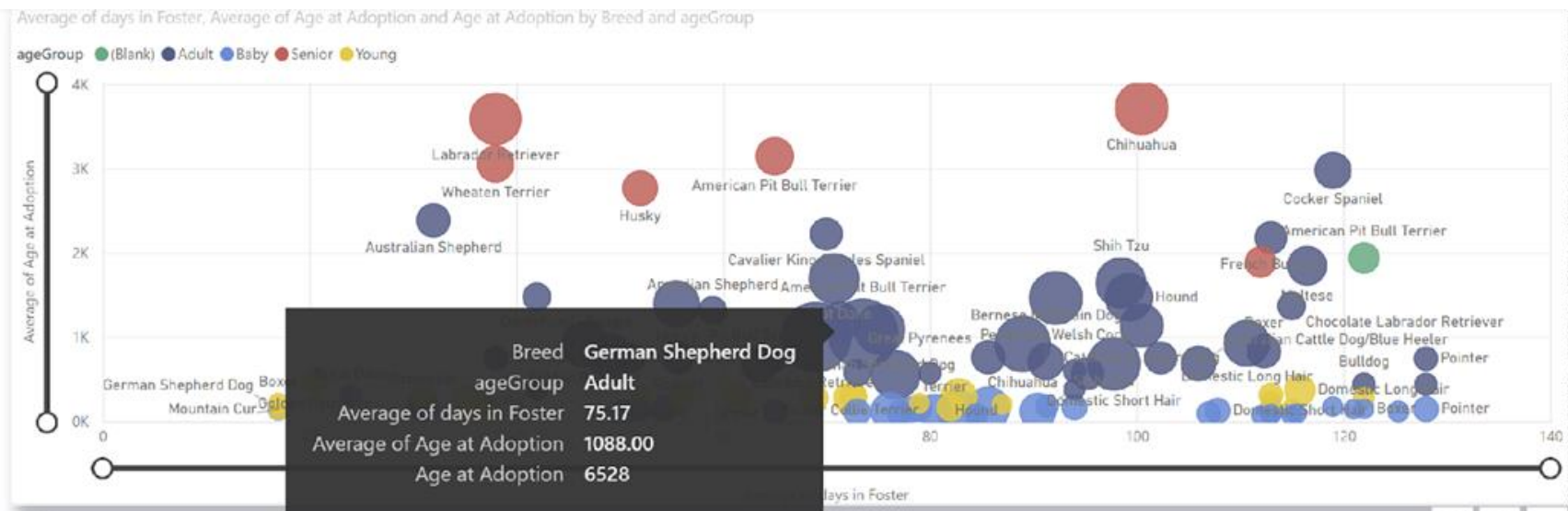
# Current Dashboard

- Status
- Adopted
  - Available
  - Euthanized
  - Hold
  - Intake

- Species, Breed
- Cat
  - Dog

- Receive Date
- 2022
  - 2021
  - 2020
  - 2019
  - 2018
  - 2017
  - 2016
  - 2015

- Contact Location
- (Blank)
  - M
  - 11
  - 24
  - 55
  - AB
  - AE
  - AK

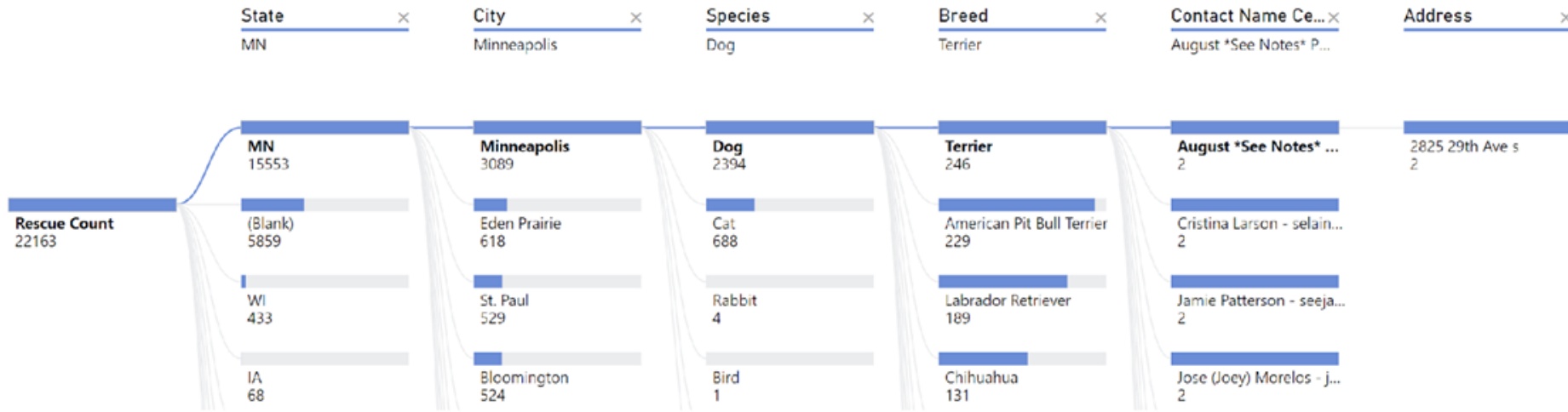


- Visualizations
- Fields

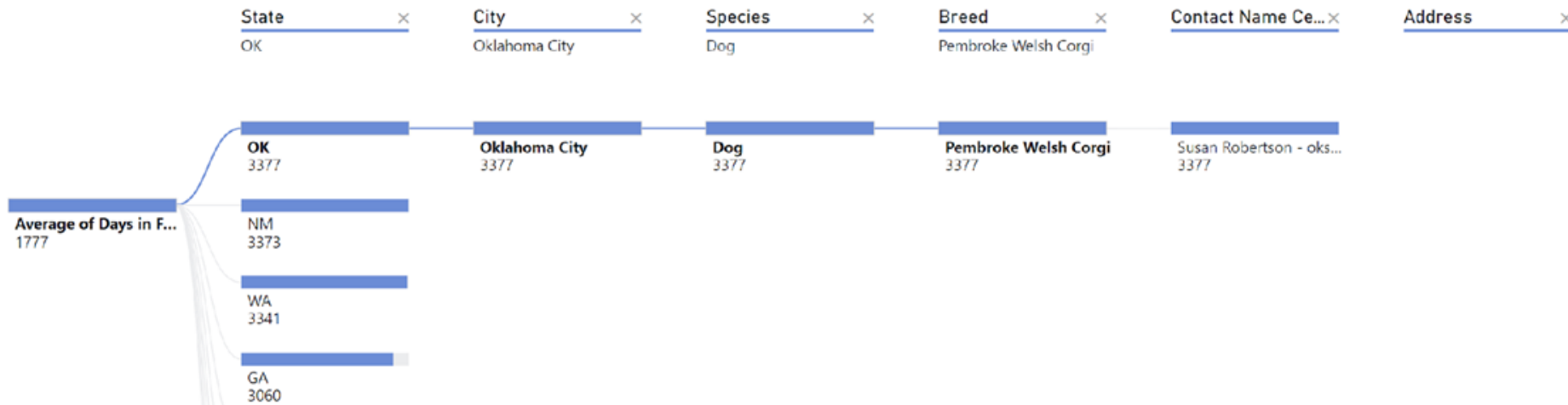


# Current Dashboard

Rescue Counts by Location, Species, Breed and Contact



Average Days in Foster by Location, Species, Breed and Contact

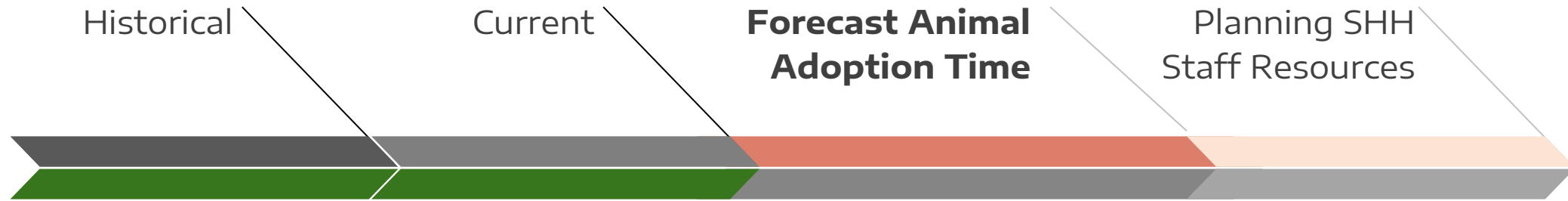


- Fields
- Visualizations
- Filters

Andrei Jula

Python Integration  
with MS PowerBI & Google Colab

# Animal Rescue as a Nonprofit Organization



## Descriptive

Self sufficient, automatic data reloads supporting dashboards in MS PowerBI

### What happened?

Historical SHH Animal information, Veterinary appointments, Fosters and Adopters

### What is happening?

Current Animal Intakes to be Fostered or Adopted

## Predictive

Manually generated models and provided insights

### What will happen?

Is the intake to foster or adoption time longer?

### Why will it happen?

Are the animal size, location & age at adoption influencing the adoption time?

## Prescriptive

Nice to have but not there yet

### What should I do?

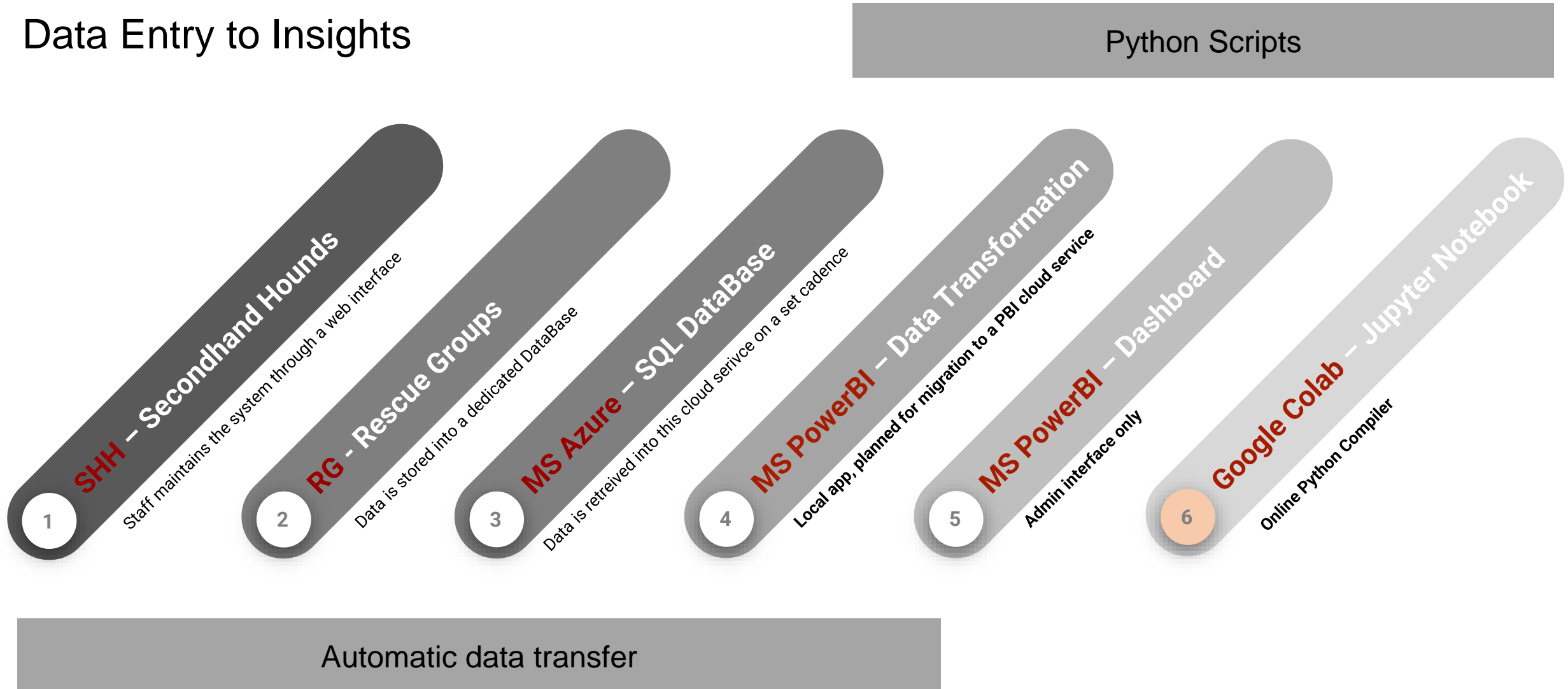
Plan for additional staff

### Why should I do it?

Targeted possible adopter reach outs

# Secondhand Hound Data Roadmap

## Data Entry to Insights



# Python for BI Environments



## Pros

- Self sufficient, interactive and user friendly dashboards & reports
- Python visuals can use any Python packages, including custom

## Integrations

- MS Visual Studio & Visual Code

## Limitations

- Pandas data frames only
- Supported PBI service packages: matplotlib, numpy, pandas, scikit-learn, scipy, seaborn, statsmodels, xgboost
- Maximum 150K rows / 250MB / 5 mins processing
- Charts are not interactive, at 72DPI only
- No column renaming



## Pros

- Web service for Jupyter Notebooks
- Main Python libraries already installed and up to date
- Well suited for Machine Learning and data analysis

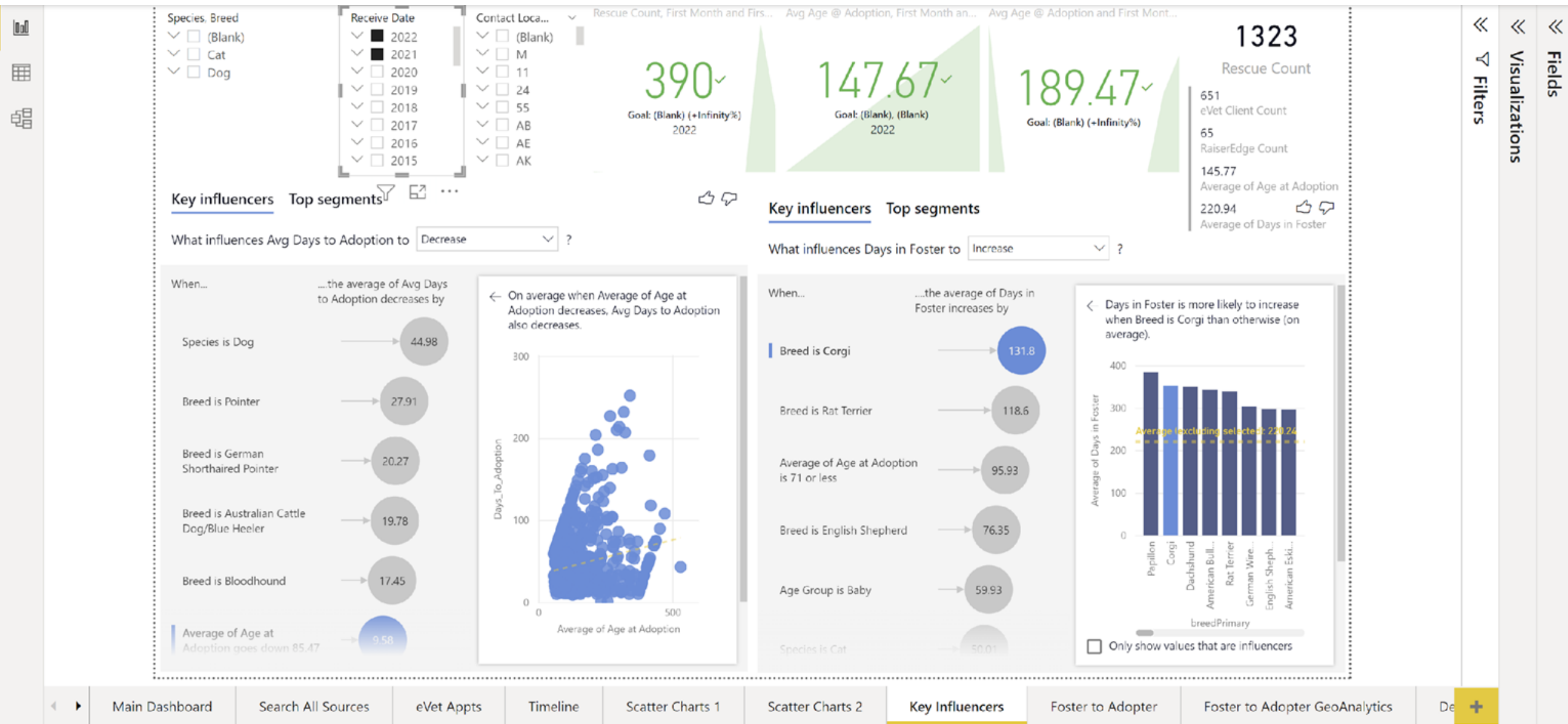
## Integrations

- No need for additional code parsers

## Limitations

- Not self sufficient for nontechnical business users
- An environment catered for data analysts, requiring custom coding and interpretation

# Current Dashboard



# Data Preprocessing

**Imputation** – fill in the missing values, without affecting the model outputs

Correlation Heatmaps – helping identifying which columns cannot be grouped in our tests

Histograms – provide an insight on how data is skewed

Consolidate the date-time information into a single date-time so that we can use it as an index in Pandas

- Define SHH Class Specific Variables

- Impute Missing Data

- Down-sample and Normalize data

- Split Data for Training and Testing



# Choosing The Right Data Models

## **Linear & Polynomial Regression**

Regression analysis on relying on dependent and independent variables / predictor and response

## **Decision Trees using Gini & Entropy – Supervised Machine Learning**

### **Classification Trees**

A classification tree is an algorithm where the target variable is categorical, splitting into classes that belong to the response variables such as Yes or No

### **Regression Trees**

A Regression tree is an algorithm where the target variable is continuous.

## **Simple Neural Network**

Flexible, with predictions for binary, numeric and categorical outcomes

Great performance for unstructured data, without the need for linear relationships, handling well outliers

## **Recurrent Neural Network Long-Short Term Memory Architecture – Deep Learning**

Great at Modeling Sequenced / Time Series Data

Effective at pattern learning

# Python Scripts in MS PowerBI

- Python scripts can be inserted in the Transform Data module of MS PowerBI
- The script is ran during the data reload step
- The generated table and custom fields are available for the data model

The screenshot displays the Power Query Editor interface. The ribbon at the top includes tabs for File, Home, Transform, Add Column, View, Tools, and Help. The 'Transform' tab is active, showing various data manipulation options like 'Choose Columns', 'Remove Columns', 'Keep Rows', 'Remove Rows', 'Sort', 'Split Column', 'Group By', 'Data Type: Text', 'Use First Row as Headers', 'Replace Values', 'Merge Queries', 'Append Queries', and 'Combine Files'. On the left, the 'Queries' pane lists several queries, with 'SecondHandHounds Animals' selected. The central area shows a Python script being executed, with a preview table below it. The script is as follows:

```
= Python.Execute("# 'dataset' holds the input data for this script#\n#\nimport pandas as pd#\n#\ndf = pd.DataFrame(dataset['adoptionFeeString'])#\n#\ndataset['adoptionFeeString'] = pd.to_numeric(df['adoptionFeeString'], errors='coerce')#\n#\ndf = df.bfill(axis=None, inplace=False, limit=None, downcast='infer')#\n#\nprint(df)#\n#\n",[dataset=#\"Renamed Columns\"]")
```

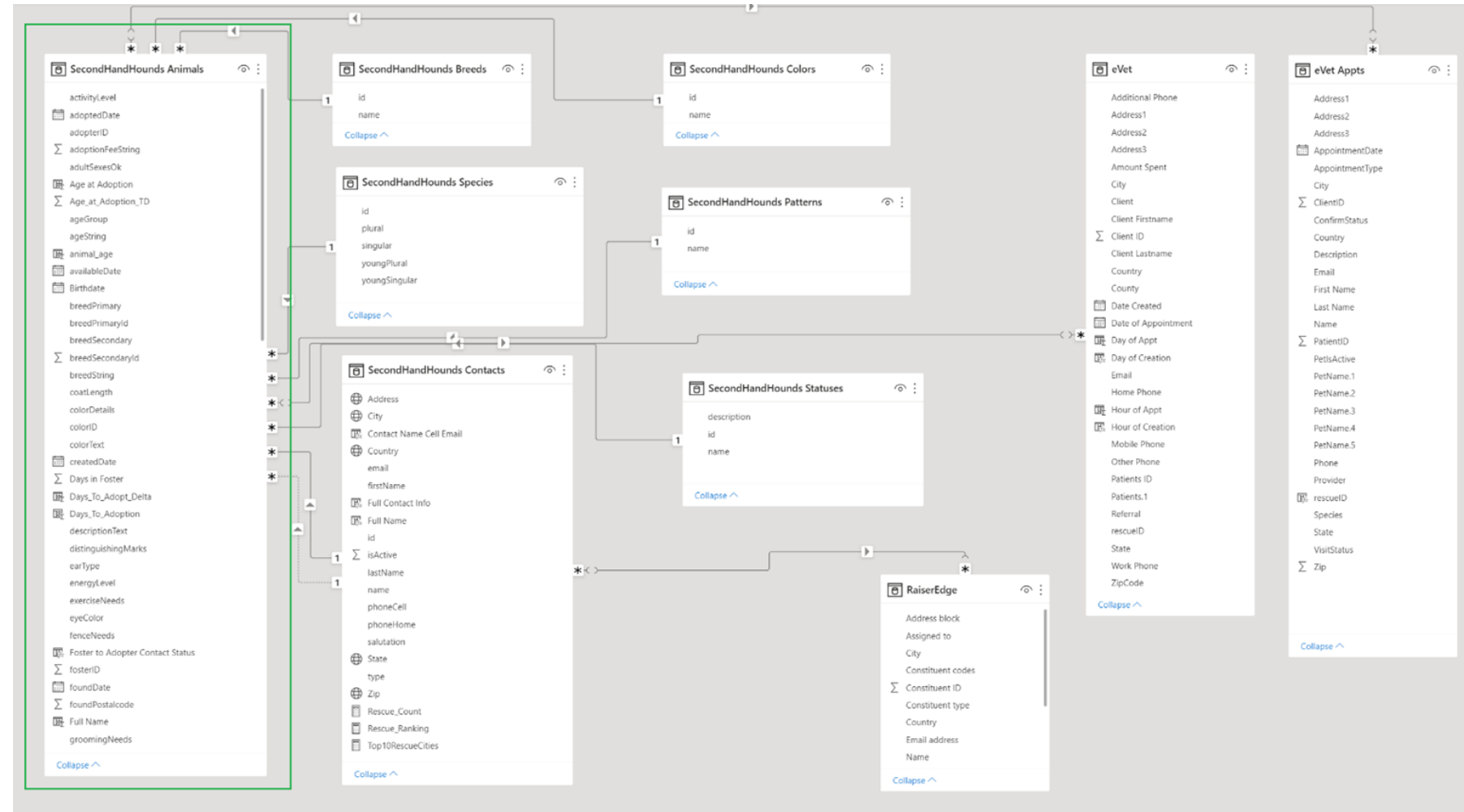
The preview table shows two columns: 'Name' and 'Value'. The first row has 'dataset' as the name and 'Table' as the value. The second row has 'df' as the name and 'Table' as the value.

On the right, the 'Query Settings' pane is open, showing the 'PROPERTIES' section with the query name 'SecondHandHounds Animals' and the 'APPLIED STEPS' section. The applied steps list includes: Source, Navigation, Check Adopted Date, Remove Empty Created Dates, Remove Empty Receive Dates, Changed Type Date to Text fo..., Renamed Columns, Run Python script - BackFill &..., Expanded Value, Undo Python Data Type Chan..., Birth Adopt Receive Dates, Age At Adoption, Changed Type to DateTime, Changed Type to Date, Column Rename for Python C..., Remove Columns, Filtered Blank ID Rows, Rename Column, and Sorted Rows.

At the bottom of the window, it says '2 COLUMNS, 2 ROWS Column profiling based on top 1000 rows' and 'PREVIEW DOWNLOADED ON NOVEMBER 5, 2022'.

# Python Scripts in MS PowerBI

- The generated tables and custom fields are available for the data model



# Python Dashboard Objects in MS PowerBI

Receive Date

- 2022
- 2021
- 2020
- 2019
- 2018
- 2017

Breed

- Abyssinian
- Affenpinscher
- Afghan Hound
- African
- African Grey
- Airedale Terrier
- Akbash
- Akhal-Teke
- Akita
- Alaskan Klee Kai
- Alaskan Malamute
- Alpine
- Amazon
- American
- American Bulldog

Status

- Treatment
- Passed
- Away
- Not Available
- Intake
- Hold
- Euthanized

Age Group

- Young
- Senior
- Baby
- Adult

Species

- Bird
- Cat
- Chinchilla
- Dog
- Ferret

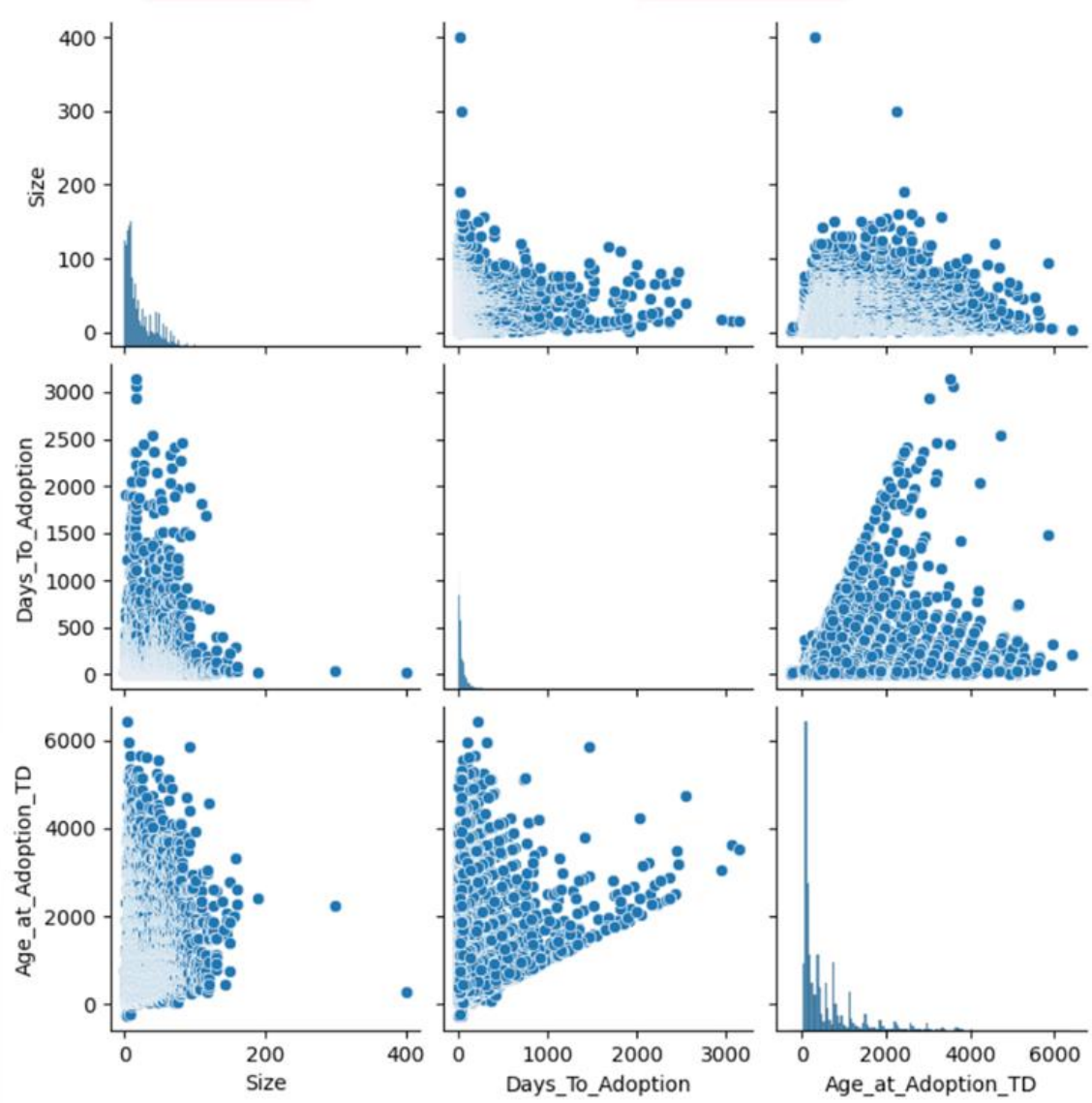
Size

- Large
- Medium
- Small
- X-Large

22,432  
SHH Rescues

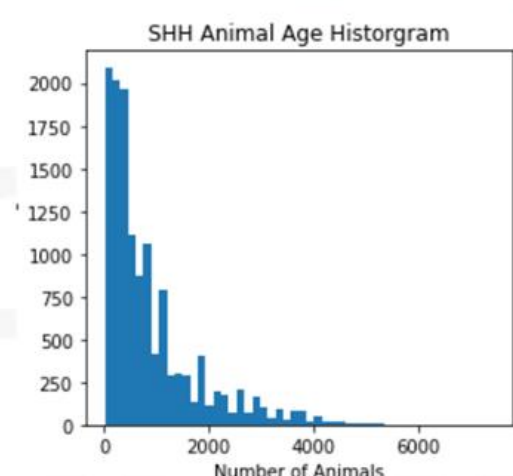
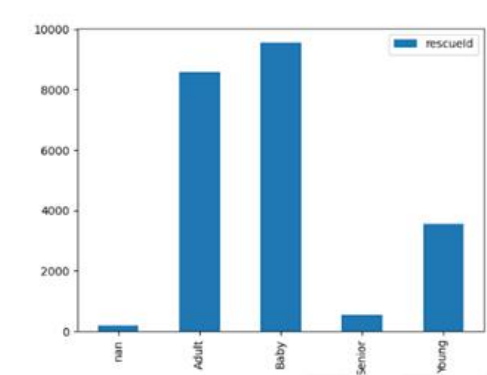
Age at Adoption and Days\_To\_Adoption

sex, Size, Days\_To\_Adoption, Age at Adoption, Age\_at\_Adoption\_TD, animal\_age, Days in Foster, statusText and Species



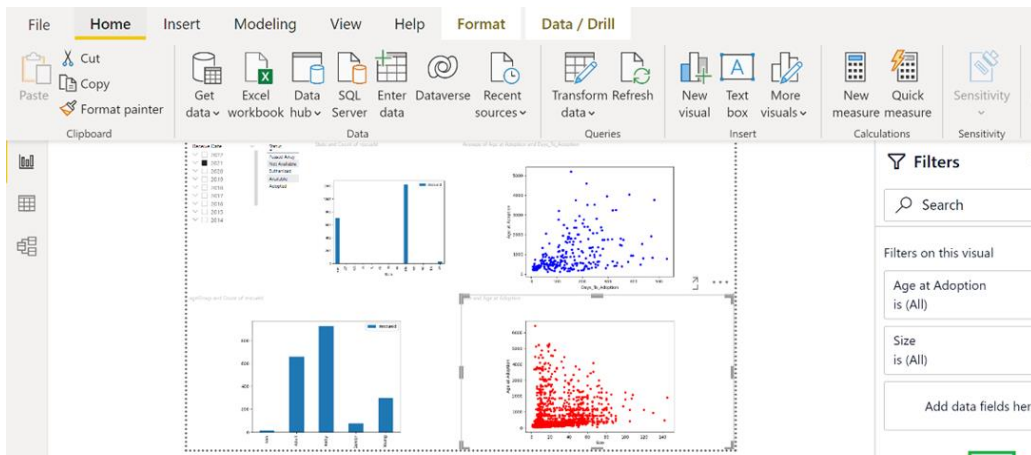
ageGroup and Count of rescued

Age at Adoption and Size



# MS PowerBI Python Objects – Interpretation in Visual Studio Code

- MS PowerBI can be integrated with MS Visual Studio or MS Visual Code for a friendlier parser environment



```
PythonEditorWrapper.PY - Visual Studio Code
PythonEditorWrapper.PY C:\PythonEditorWrapper_1baf421d-84d6-44bf-bd8b-e2b6284772cc X Release Notes: 1.73.0 PythonEditorWrapper.PY C:\PythonEditorWrapper_b156a6f3-9ae5-4635-8860-8d4...
C:\Users> ajula > PythonEditorWrapper_1baf421d-84d6-44bf-bd8b-e2b6284772cc > PythonEditorWrapper.PY > ...
1 # Prolog - Auto Generated #
2 import os, uuid, matplotlib
3 matplotlib.use("Agg")
4 import matplotlib.pyplot
5 import pandas
6
7 import sys
8 sys.tracebacklimit = 0
9
10 os.chdir(u'C:/Users/ajula/PythonEditorWrapper_1baf421d-84d6-44bf-bd8b-e2b6284772cc')
11 dataset = pandas.read_csv('input_df_beccd07b-5d0d-42aa-adff-3c846138ab76.csv')
12
13 matplotlib.pyplot.figure(figsize=(5.555555555555556,4.166666666666667), dpi=72)
14 matplotlib.pyplot.show = lambda args=None,kw=None: matplotlib.pyplot.savefig(str(uuid.uuid1()))
15 # Original Script. Please update your script content here and once completed copy below section back to the original editing window #
16 # The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:
17
18 # dataset = pandas.DataFrame(ageGroup, rescueId)
19 # dataset = dataset.drop_duplicates()
20
21 # Paste or type your script code here:
22 import matplotlib.pyplot as plt
23 dataset.plot(kind='scatter', x='Size', y='Age at Adoption', color='red')
24 plt.show()
25
26 # Epilog - Auto Generated #
27 os.chdir(u'C:/Users/ajula/PythonEditorWrapper_1baf421d-84d6-44bf-bd8b-e2b6284772cc')
28
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS C:\Users\ajula> & C:/Users/ajula/anaconda3/python.exe c:/Users/ajula/PythonEditorWrapper_1c29a03e-0014-4a67-be62-69ec9e5fe99b/PythonEditorWrapp...
FileNotFoundError: [WinError 2] The system cannot find the file specified: 'C:/Users/ajula/PythonEditorWrapper_b156a6f3-9ae5-4635-8860-8dac65db973c...'
PS C:\Users\ajula>
```

```
Python script editor
Duplicate rows will be removed from the data.
1 # The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:
2
3 # dataset = pandas.DataFrame(ageGroup, rescueId)
4 # dataset = dataset.drop_duplicates()
5
6 # Paste or type your script code here:
7 import matplotlib.pyplot as plt
8 dataset.plot(kind='scatter', x='Size', y='Age at Adoption', color='red')
9 plt.show()
```

The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The main editor window displays a Python script for data processing and visualization. The bottom status bar shows 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8 with BOM', 'CRLF', 'Python', and '3.9.13 (base: conda)'. A 'Python script editor' window is open at the bottom, showing a Python script for data processing and visualization. The script includes a preamble for removing duplicate rows and a main script for plotting a scatter plot. The 'Python script editor' window also shows a warning message: 'Duplicate rows will be removed from the data.'



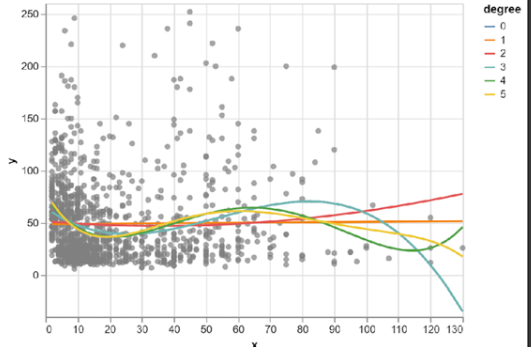
# Python Scripts in Google Colab

6940 SHH - Data Prep - O2 Polynomial Regression.ipynb ☆

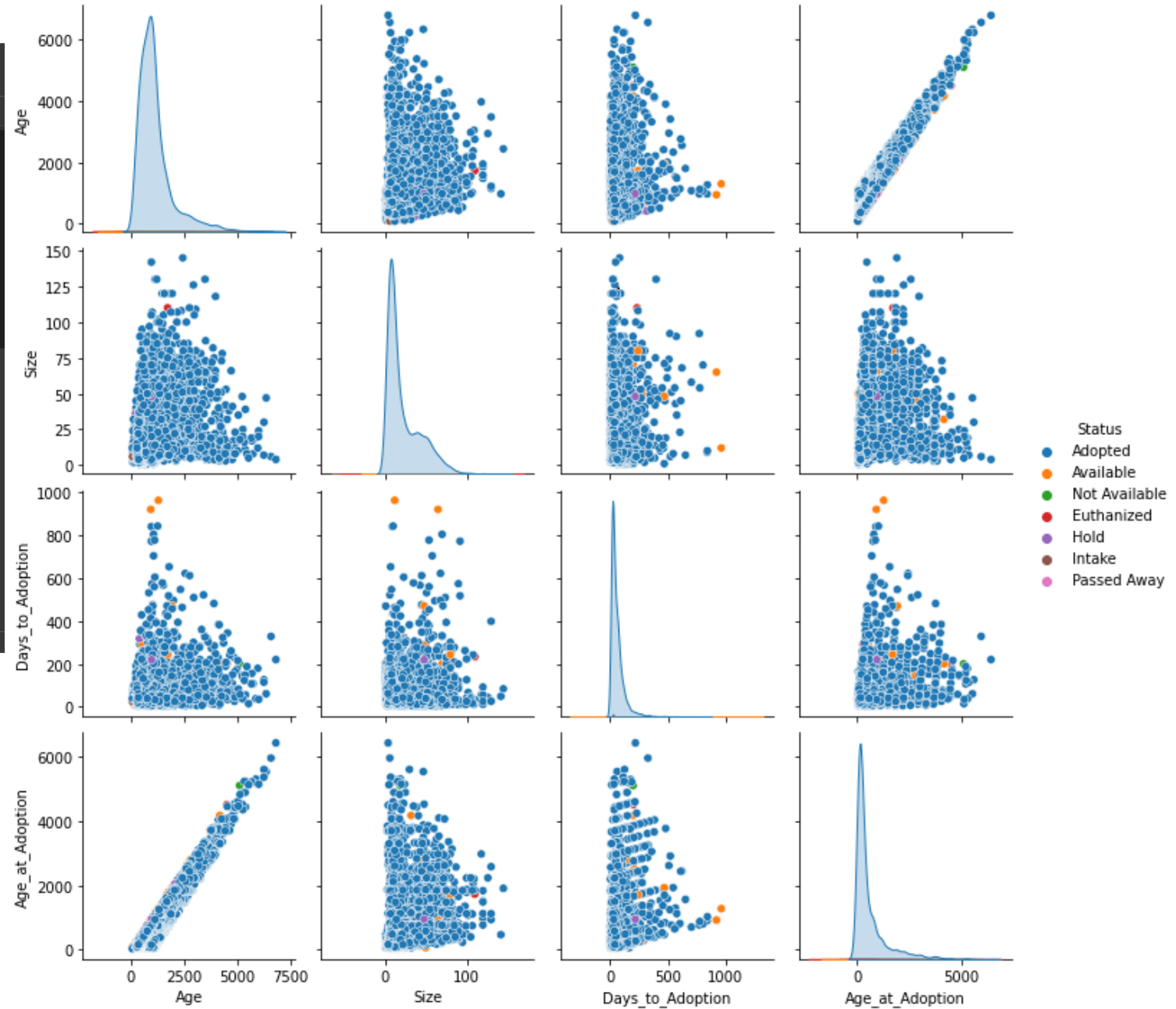
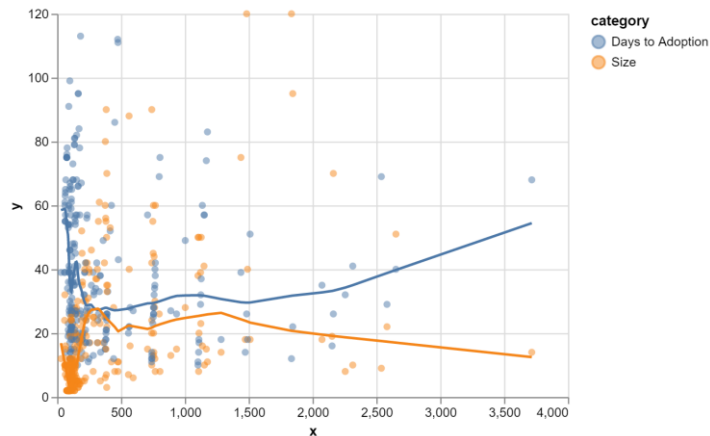
File Edit View Insert Runtime Tools Help

+ Code + Text

```
1 x = shh.Size
2 y = shh.Days_to_Adoption
3 source = pd.DataFrame({"x": x, "y": y})
4 degree_list = [0, 1, 2, 3, 4, 5] # Define the degree of the polynomial fits
5 base = alt.Chart(source).mark_circle(color="gray").encode(
6     alt.X("x"), alt.Y("y"))
7 polynomial_fit = [base.transform_regression(
8     "x", "y", method="poly", order=order, as_=["x", str(order)])
9     .mark_line()
10    .transform_fold([str(order)], as_=["degree", "y"])
11    .encode(alt.Color("degree:N"))
12    for order in degree_list]
13 alt.layer(base, *polynomial_fit)
```

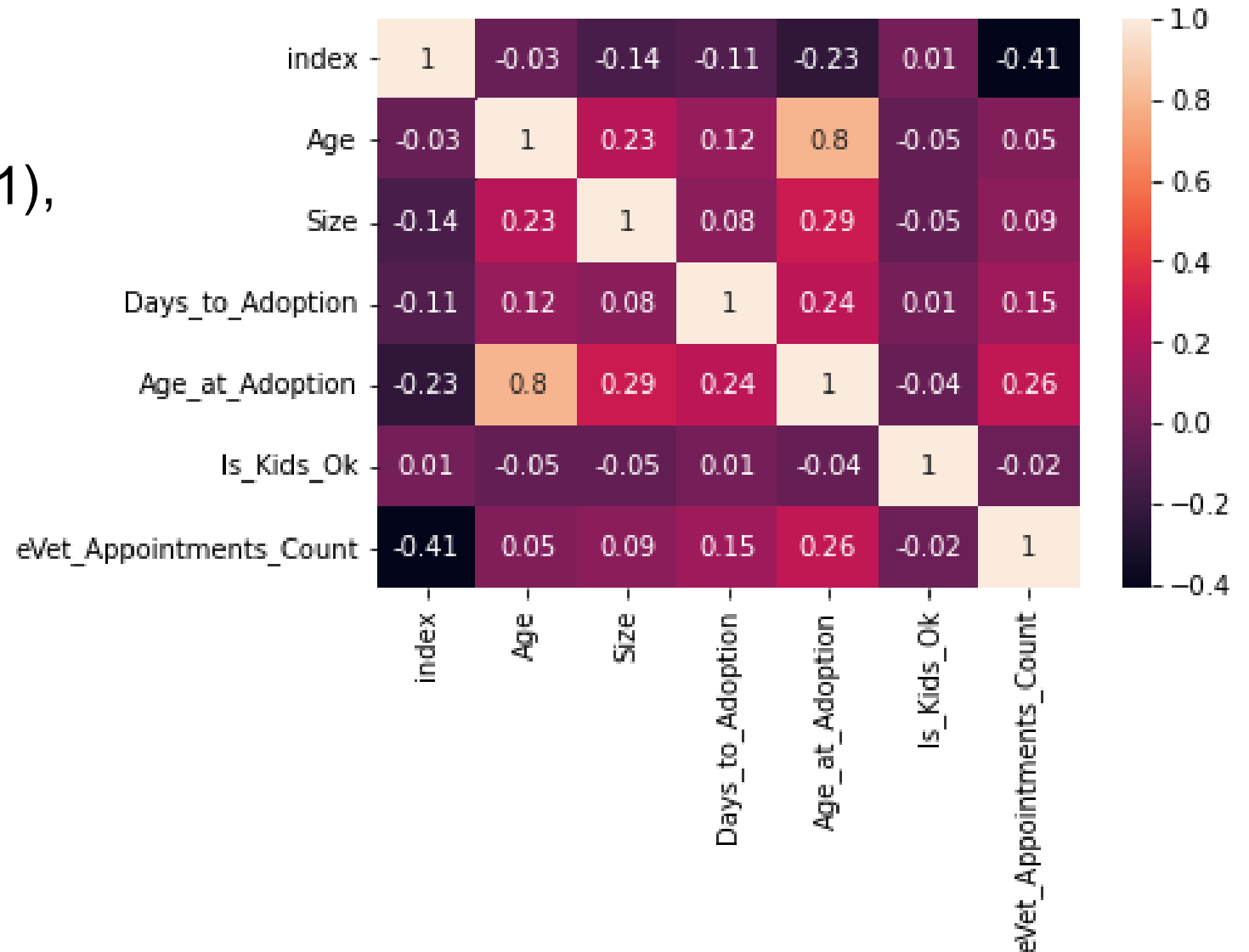


0s completed at 5:51 PM



# SHH Data Heat Map

- The closer the value is to 1 (or -1), the stronger a relationship
- The closer a number is to 0, the weaker the relationship
- Negative relationships imply the decrease of a value, while the other increases



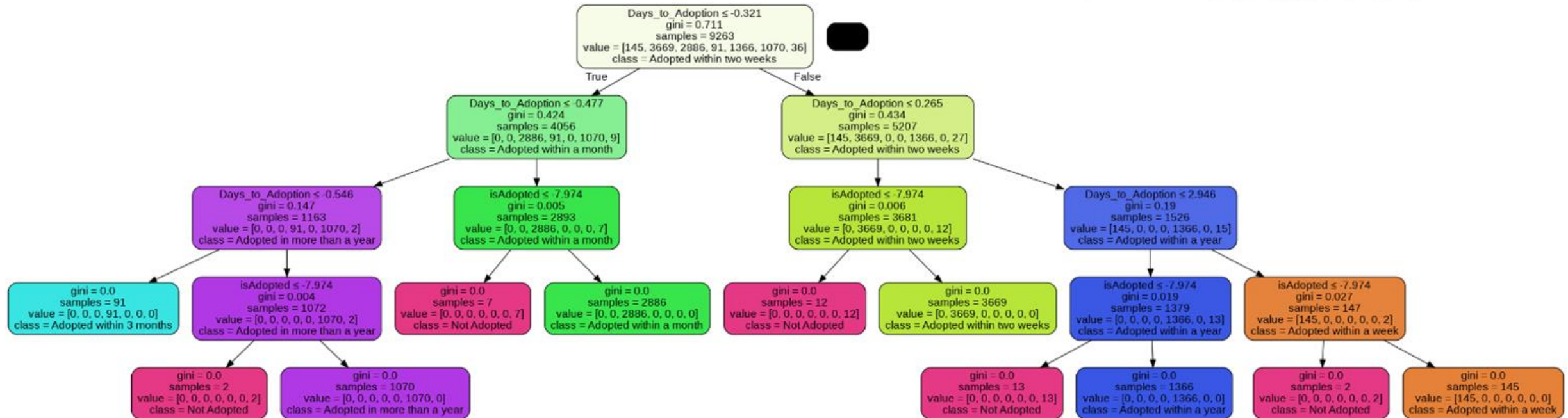
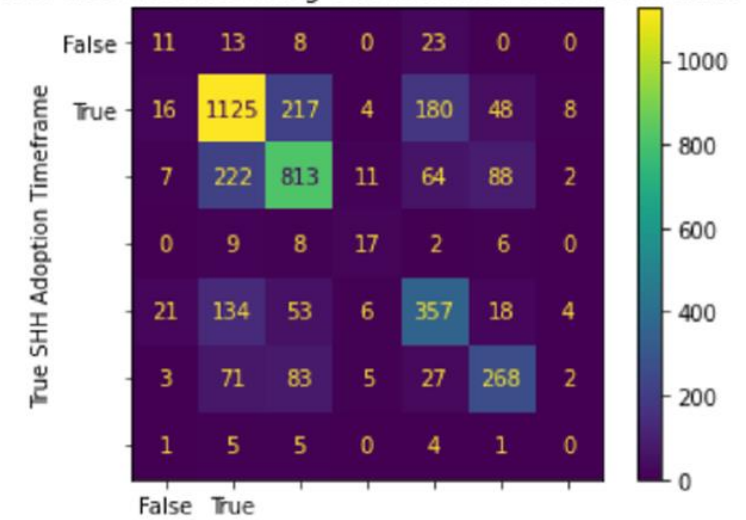


# Decision Trees – Supervised Predictive Analytics

## Confusion Matrix

- Accuracy - how often the model is correct: 82.90
- Precision - how well the model made its predictions: 72.16
- Sensitivity / Recall - how well the model is predicting positives: 82.90
- Specificity - how well the model is predicting negatives: 82.90
- F-Score - harmonic mean of precision and sensitivity: 75.29

SHH Tree Classifier using Gini Predicted Confusion Matrix



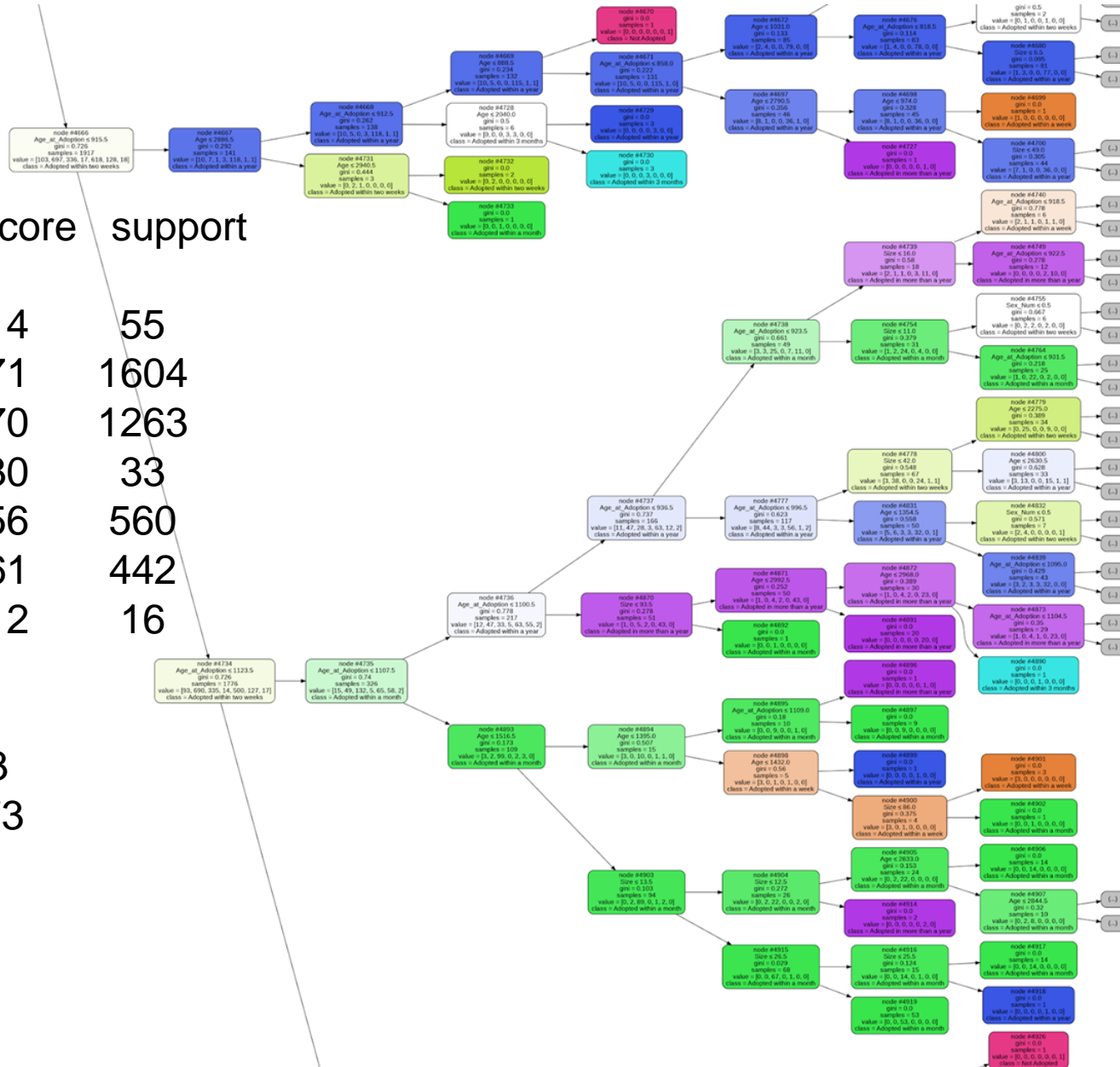
# Python Scripts in MS PowerBI

Report :

Adopted in more than a year  
 Adopted within 3 months  
 Adopted within a month  
 Adopted within a week  
 Adopted within a year  
 Adopted within two weeks  
 Not Adopted

	precision	recall	f1-score	support
Adopted in more than a year	0.13	0.15	0.14	55
Adopted within 3 months	0.72	0.70	0.71	1604
Adopted within a month	0.71	0.70	0.70	1263
Adopted within a week	0.26	0.36	0.30	33
Adopted within a year	0.53	0.59	0.56	560
Adopted within two weeks	0.61	0.60	0.61	442
Not Adopted	0.11	0.12	0.12	16

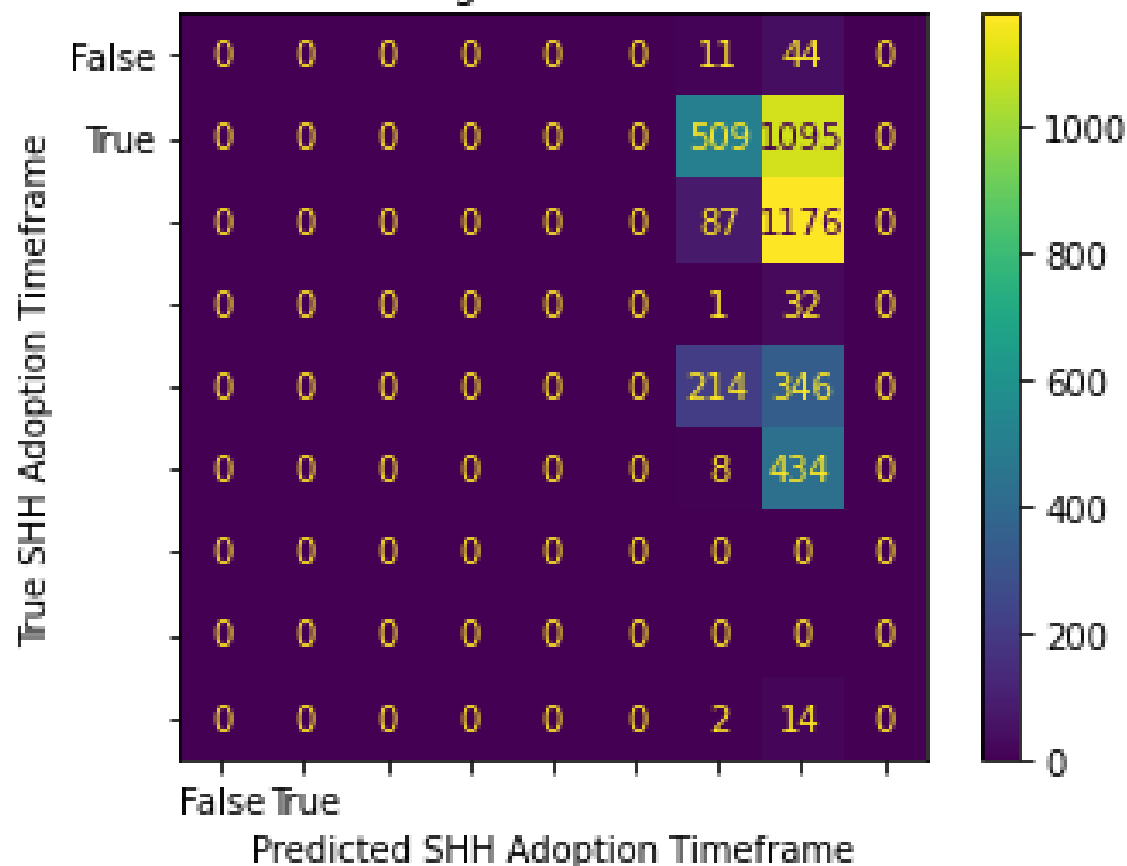
accuracy	0.66	3973		
macro avg	0.44	0.46	0.45	3973
weighted avg	0.66	0.66	0.66	3973



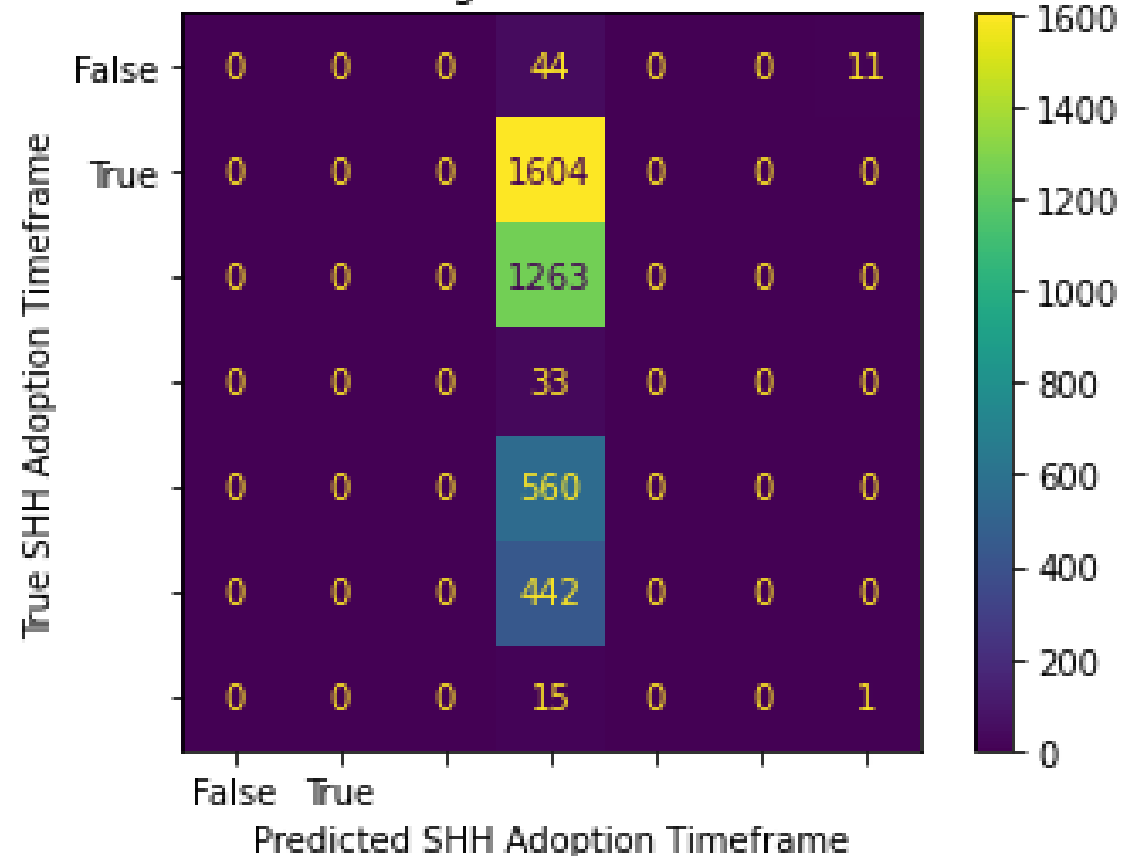
# Confusion Matrix from a saved Model

Confusion Matrix from a saved Model automatically dropped null rows, repositioned the true vs false matrix and added the values within the same row, without affecting the ability to predict

SHH Tree Classifier using Gini Predicted Confusion Matrix



SHH Tree Classifier using Gini Predicted Confusion Matrix

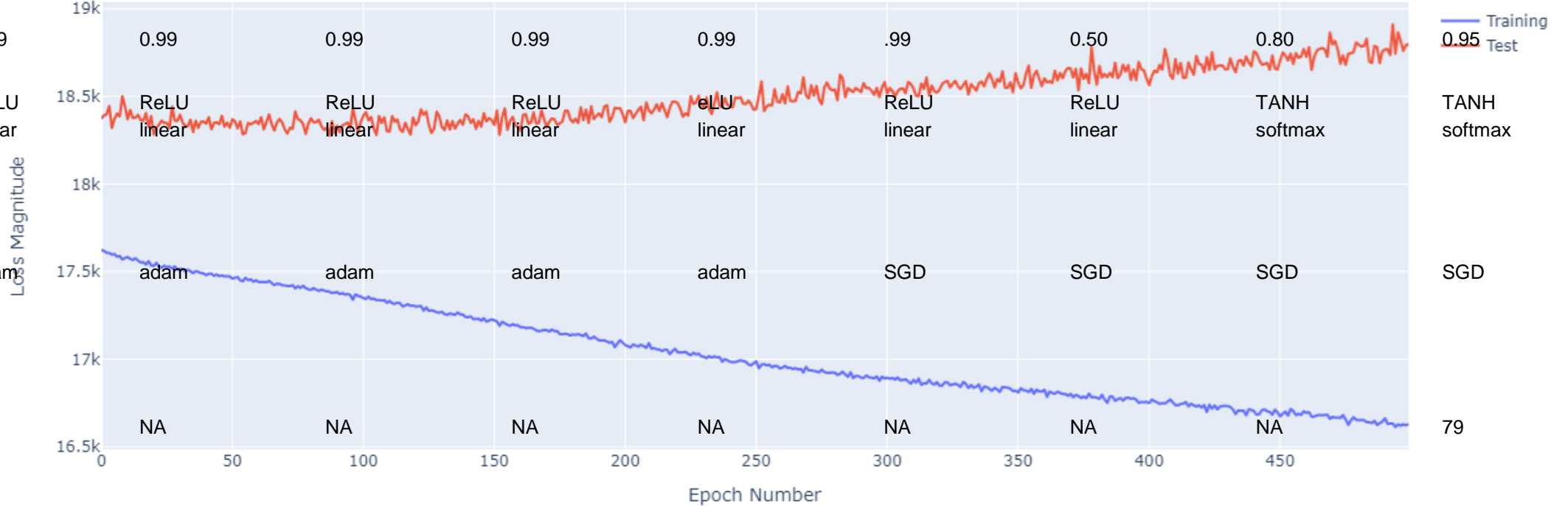


# ML Simple Neural Network

```
6940 SHH - 05 - ML Simple Neural Network.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
False
↑ ↓ ↻ ⌨ ⚙ 📄 🗑 ⋮
1 print('Defining compiling parameters for tweaking the model to its best (smallest) loss')
2 model = Sequential()
3
4 # Model compiling parameters
5
6 activation_hidden = 'relu' #'elu' 'tanh'
7 activation_output = 'linear' #'softmax'
8
9 print('Defining the hidden dense layers, one or many, with different number of neurons, using different activation functions')
10 model.add (Dense(32, input_shape=(6,), activation=activation_hidden, name='dense_1'))
11 model.add (Dense(16, activation=activation_hidden, name='dense_2'))
12 #model.add (Dense(32, activation=activation_hidden, name='dense_3'))
13
14 print('Defining the output dense layer as a linear or softmax activation')
15 model.add (Dense(1, activation = activation_output, name = 'output'))
16
17 print('We are using the adam (Adaptive Moment Estimation) optimizer as baseline')
18 #optimizer='adam' #'rmsprop'
19 optimizer = tf.keras.optimizers.Adam(lr = 0.0001)
20
21 # Learning Rate and Momentum variables used in the SGD optimizer only
22 lr = 0.0001
23 momentum = 0.99
24 #optimizer = SGD(lr=lr, momentum=momentum)
25
26 print('In order to know if the model is properly learning, we use a mean squared error ('mse') loss function')
27 loss='mse'
28 #loss='categorical_crossentropy'
29
30 print('To report the performance of it we will adopt the mean average error ('mae') metric')
31 #metrics='accuracy'
32 metrics='mae'
33 #metrics='mse'
34 #metrics='mape'
35
36 model.compile(optimizer, loss, metrics)
37
38 print('Using the summary method from Keras, we are printing the model structure')
39 model.summary()
```

# Simple Neural Network

Number of neurons in the hidden layer(s)	Layer 1 = 16	Layer 1 = 16 Layer 2 = 16	Layer 1 = 32 Layer 2 = 16 Layer 3 = 16	Layer 1 = 16 Layer 2 = 16 Layer 3 = 32	Layer 1 = 128 Layer 2 = 64 Layer 3 = 32	Layer 1 = 64 Layer 2 = 32	Layer 1 = 32 Layer 2 = 16 Layer 3 = 32	Layer 1 = 128 Layer 2 = 64 Layer 3 = 32	Layer 1 = 128 Layer 2 = 64 Layer 3 = 32
Learning Rate	0.001	0.001	0.001	0.00001	0.001	0.0001	0.001	0.001	0.001
Momentum	0.99	0.99	0.99	0.99	0.99	0.99	0.50	0.80	0.95
Activation Functions Hidden / Output	ReLU linear	ReLU linear	ReLU linear	ReLU linear	ReLU linear	ReLU linear	ReLU linear	TANH softmax	TANH softmax
Optimizers	adam	adam	adam	adam	adam	SGD	SGD	SGD	SGD
Early Stop Epoch Count Patience as 10	NA	NA	NA	NA	NA	NA	NA	NA	79
Time in seconds	125	134	114	143	132	143	145	145	223
Loss as MSE/MAE	loss: 17885.29 mae: 55.53	loss: 17924.82 mae: 55.88	loss: 17941.82 mae: 55.74	loss: 18515.61 mae: 58.81	loss: 18031.17 mae: 56.97	loss: 20105.48 mae: 58.62	loss: 18562.87 mae: 57.24	loss: 0.2194 mae: 56.17	loss: 24755.57 mae: 68.20

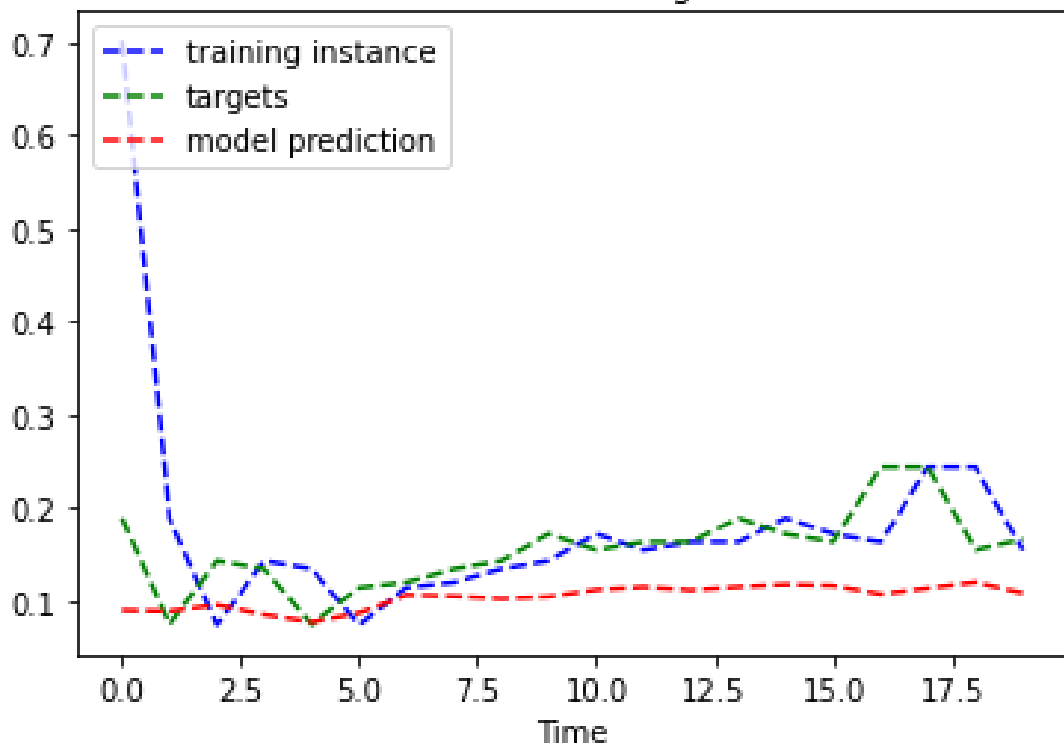


# Recurring Neural Network Long-Short Term Memory Architecture

RNN LSTM Model using \*ADAM (Adaptive Moment Estimation) as an optimizer and Mean Squared Error as loss suppression

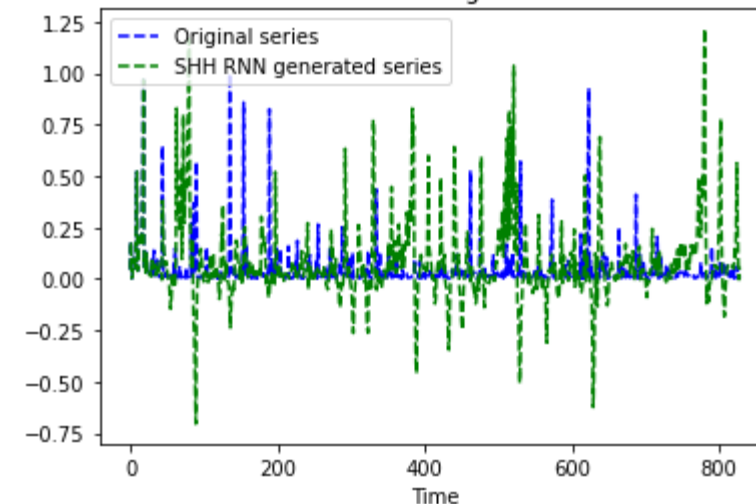
ADAM - Use of an exponentially decaying average of past gradients

Model Testing



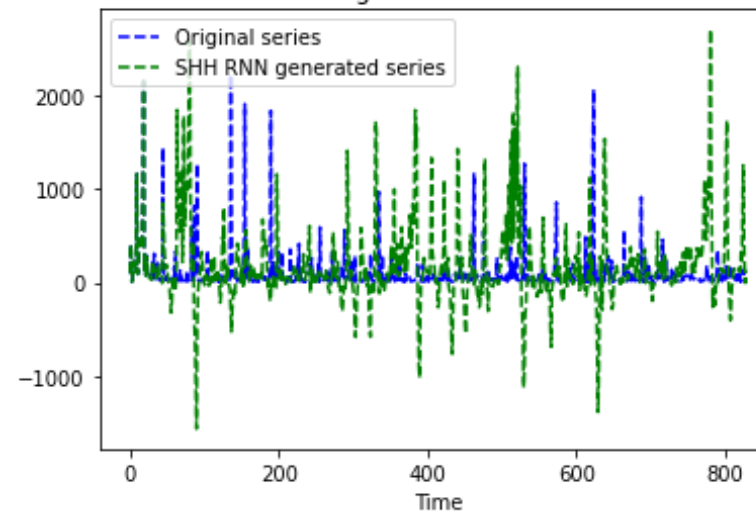
Model Plot on Predicted Sequence on SHH Data since 01/01/2020

SHH RNN vs. Original series



Plot on normalized Days to Adoption Scale

SHH RNN vs. Original series with normal scale



Plot on inversed to normal Days to Adoption Scale



# Resources / Bibliography

[2023 Great Lakes Data & Analytics Summit - Call for Speakers \(witinc.com\)](https://www.witinc.com)

## Organization Info

1. SHH Organization retrieved from [Secondhand Hounds | Animal Rescue | Twin Cities, Minnesota](#)
2. SHH Contacts [retrieved from Our Team | Secondhand Hounds](#)
3. Oakland University News, School of Business Administration retrieved from [Data For Good: Project delivers outcomes to benefit animal rescue efforts \(oakland.edu\)](#)

## Environments

1. [Setting the Python/Pandas/MS Visual Studio Environment steps retrieved from Run Python Scripts in Power BI Desktop - Power BI | Microsoft Learn Environment](#)
2. [Run Python Scripts in Power BI Desktop - Power BI | Microsoft Learn](#)
3. [Create Power BI visuals using Python in Power BI Desktop - Power BI | Microsoft Learn](#)
4. [Installation - pip documentation v22.3 \(pypa.io\)](#)
5. [Python in Visual Studio Code](#) as an [Extension in Visual Studio Code](#) – Optional, but highly recommended for an intuitive parser, automatic code backup with Microsoft or GitHub, Intelligent Code Completion, variables, methods and imported modules, debugging, source control, etc.
6. [Documentation for Visual Studio Code](#)
7. [Use an external Python IDE with Power BI - Power BI | Microsoft Learn](#)
8. [The Python Tutorial — Python 3.10.8 documentation](#)
9. [Pillow · PyPI](#)
10. [How To Fix The Python Pillow Error ImportError: DLL load failed while importing \\_imaging: The specified module could not be found. \(dev2qa.com\)](#)

## Data Preparation

1. [What is One Hot Encoding and How to Do It | by Michael DeSole | Medium](#)

## Decision Trees

1. [Python Decision Tree Classification Tutorial: Scikit-Learn DecisionTreeClassifier | DataCamp](#)
2. [Decision Tree Classifier - The Click Reader](#)
3. [Decision Tree Classifier with Sklearn in Python • datagy](#)
4. [Decision Trees: Gini vs Entropy | Quantdare](#)
5. [python - Facing ValueError: Target is multiclass but average='binary' - Stack Overflow](#)
6. [CART \(Classification And Regression Tree\) in Machine Learning – GeeksforGeeks](#)
7. [Random Forest in Python. A Practical End-to-End Machine Learning... | by Will Koehrsen | Towards Data Science](#)
8. [Decision Tree Classifier explained in real-life: picking a vacation destination | by Carolina Bento | Towards Data Science](#)

## Machine Learning

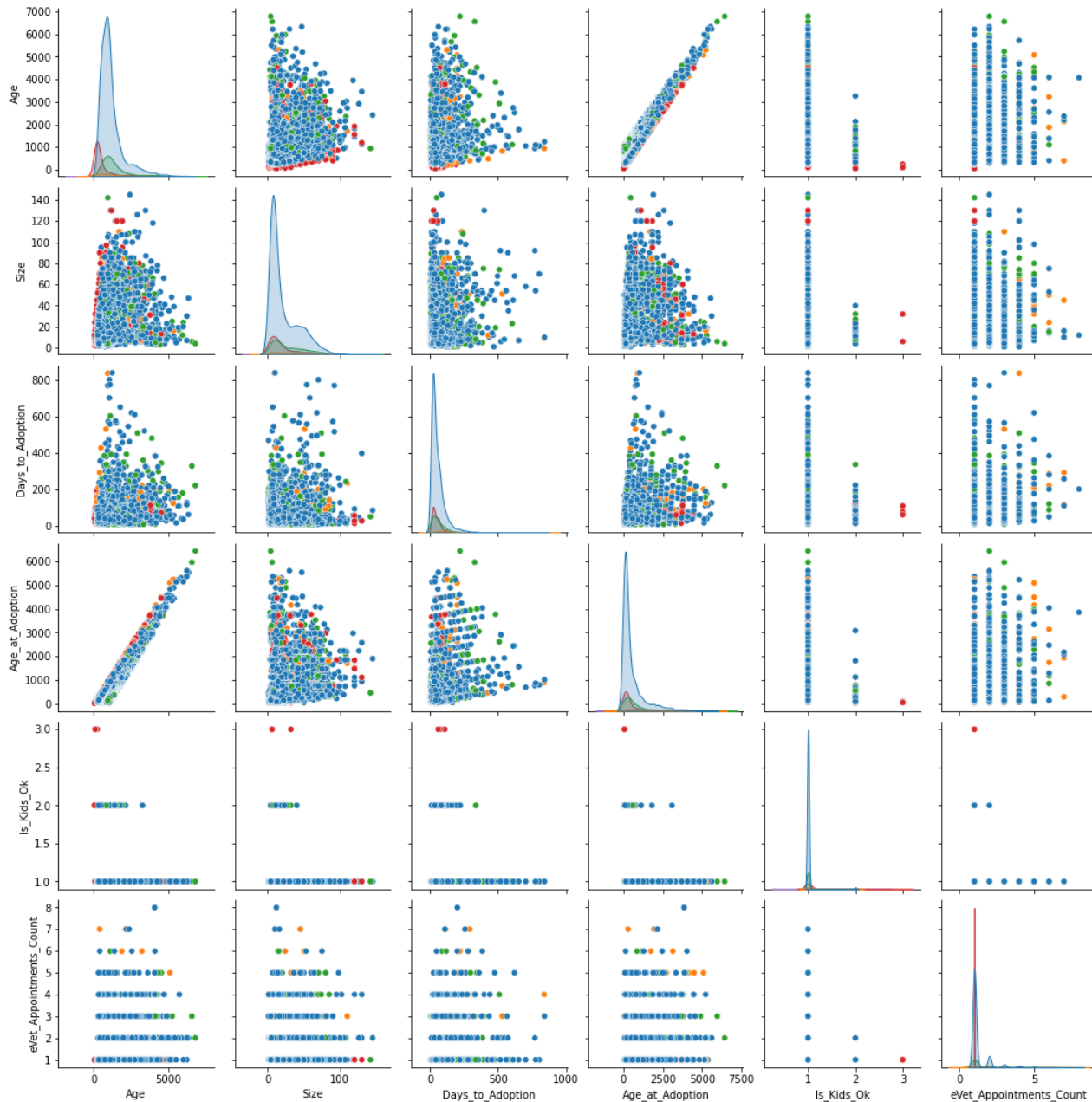
1. [Python Machine Learning - Confusion Matrix \(w3schools.com\)](#)
2. [Python Machine Learning Normal Data Distribution \(w3schools.com\)](#)
3. [Python Machine Learning Decision Tree \(w3schools.com\) -> W3Schools Tryit Editor](#)
4. [Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras - MachineLearningMastery.com](#)



# Backup



# Data Correlation – Pairplots



## Data Correlation – Pairplots

- Animal Size
- Age
- Days to Adoption
- Age at Adoption
- Is Kid Friendly
- Number of eVet Appointments

By highlighting the different statuses:

- Different Foster and Adopter
- Foster Only
- Same contact - Foster to Adopter
- No Foster or Adopter
- Adopter Only



# Python in Google Colab, new logical columns

## ▼ Create New SHH Logical Columns

```
1 # Create a new column True / False that represents the animal adoption status as binary
2 # Create additional SHH adoption segments in logical increments
3
4 shh['Is_Kids_Ok_TF'] = np.where(shh['Is_Kids_Ok'] == 1, True, False)
5
6 shh['isAdopted'] = np.where(shh['Status'] == 'Adopted', True, False)
7
8 shh['isAdopted_7'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] <= 7), True, False)
9 shh['isAdopted_14'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 7) & (shh['Days_to_Adoption'] <= 14), True, False)
10 shh['isAdopted_30'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 14) & (shh['Days_to_Adoption'] <= 30), True, False)
11 shh['isAdopted_90'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 30) & (shh['Days_to_Adoption'] <= 90), True, False)
12 shh['isAdopted_365'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 90) & (shh['Days_to_Adoption'] <= 365), True, False)
13
14 shh['isAdopted_7_365'] = np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] <= 7), 'Adopted within a week',
15 np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 7) & (shh['Days_to_Adoption'] <= 14), 'Adopted within two weeks',
16 np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 14) & (shh['Days_to_Adoption'] <= 30), 'Adopted within a month',
17 np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 30) & (shh['Days_to_Adoption'] <= 90), 'Adopted within 3 months',
18 np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 90) & (shh['Days_to_Adoption'] <= 365), 'Adopted within a year',
19 np.where((shh['Status'] == 'Adopted') & (shh['Days_to_Adoption'] > 365), 'Adopted in more than a year',
20 'Not Adopted'))))))) # catch all timeframes
21
```

# Data Prep – Understanding SHH Data, Animal Age Histogram

6940 SHH - Data Prep - 00 Understanding SHH DataSet.ipynb ☆

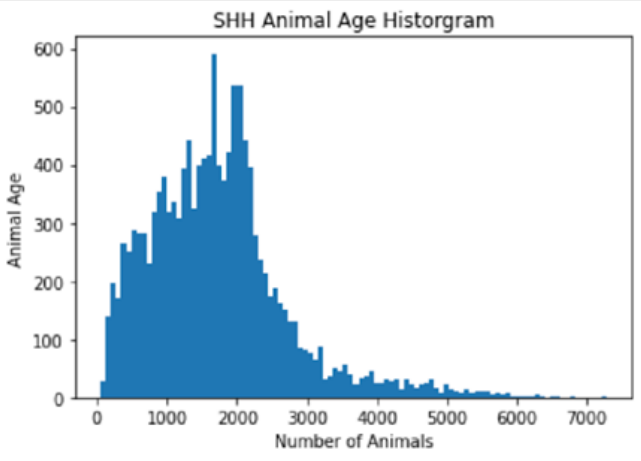
File Edit View Insert Runtime Tools Help [Last edited on November 24](#)

Comment Share Settings Profile

+ Code + Text Connect Editing

```
7
8 # Second Hand Hound Animal Age at Adoption histogram with 100 bars
9 plt.hist(shh.Age_at_Adoption, 100)
10 plt.title("SHH Animal Age at Adoption Histogram")
11 plt.xlabel("Number of Animals")
12 plt.ylabel("Animal Age at Adoption")
13 plt.show()
14
15 # Second Hand Hound Animal Size histogram with 100 bars
16 plt.hist(shh.Size, 100)
17 plt.title("SHH Animal Size Histogram")
18 plt.xlabel("Animal Size (Weight in Pounds)")
19 plt.ylabel("Number of Animals")
20 plt.show()
21
22 #plt.hist(shh.isAdopted_90_Num, 100)
23 #plt.show()
```

SHH Animal Age Histogram



Number of Animals	Animal Age
0	0
1000	300
2000	500
3000	200
4000	100
5000	50
6000	20
7000	10

# Polynomial Regression, Animal Age at Adoption based on Size

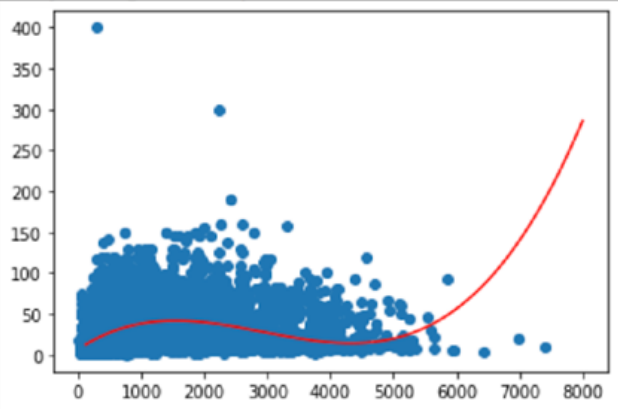
6940 SHH - Data Prep - 02 Polynomial Regression.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

```
16 print('and determine if a polynomial regression is fit for prediction')
17 print('Values close to 0 mean no relationship, close to 1, correlated')
18
19 print('R-Squared')
20 print(r2_score(y, polynomial_regression_model(x)))
21
22 print('Predicted number of days to adoption based on size:')
23 predicted_size = polynomial_regression_model(40)
24 print(predicted_size)
```

Polynomial Regression  
Study on Age at Adoption and Size



Understand the correlation between the x and y axes  
and determine if a polynomial regression is fit for prediction  
Values close to 0 mean no relationship, close to 1, correlated  
R-Squared  
0.257088123920721  
✖ Predicted number of days to adoption based on size:  
9.172285509641188

0s completed at 5:51 PM

# Decision Tree Classifier Gini – Confusion Matrix on Predicted SHH Adoption Timeframes



6940 SHH - Decision Tree Classifiers Gini & Entropy.ipynb ☆

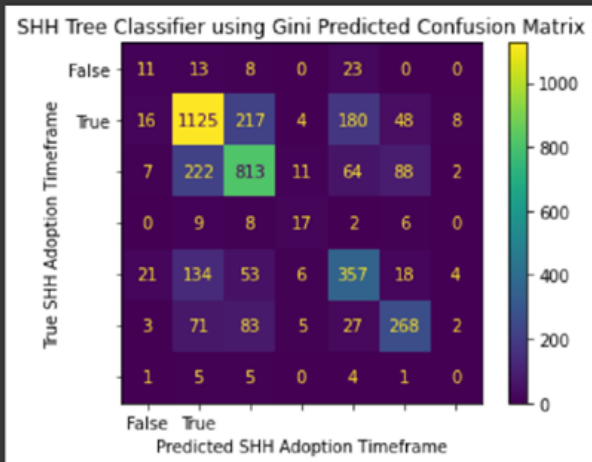
Comment Share Settings Profile

File Edit View Insert Runtime Tools Help Last saved at 2:15 PM

+ Code + Text

Connect Editing

```
[ ] 1 confusion_matrix_gini = metrics.confusion_matrix(y_test, y_pred_gini)
2
3 # Convert the gini predicted table into a confusion matrix
4 confusion_matrix_shh_adopt_gini = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix_gini, display_labels = [False, True])
5
6 # Display the confusion matrix findings
7 confusion_matrix_shh_adopt_gini.plot()
8 plt.title("SHH Tree Classifier using Gini Predicted Confusion Matrix")
9 plt.xlabel("Predicted SHH Adoption Timeframe")
10 plt.ylabel("True SHH Adoption Timeframe")
11 plt.show()
```



```
[ ] 1 # Plotting the decision boundary
```



Thank you for attending!

Hosted by:

