import pandas as pd In [354... import numpy as np from matplotlib import pyplot as plt Data Pre-processing Read Animal Centor Intakes and Outcomes data In [355... # Read intakes csv file intakes = pd.read\_csv('processed\_data/dog\_intakes\_unified.csv', dtype={ 'AnimalID': 'string', 'ID': 'string', 'Breed': 'string' 'Color': 'string', 'Gender': 'category', 'Name': 'string', 'Intake-DateTime': 'string', 'Intake-Type': 'string', 'Intake-Condition': 'string', 'Intake-Age(days)': 'int', }, parse\_dates=['Intake-DateTime']) intakes.dtypes AnimalID string[python] Out[355]: ID string[python] Breed string[python] Color string[python] Gender category Name string[python] Intake-DateTime datetime64[ns] Intake-Type string[python] Intake-Condition string[python] Intake-Age(days) int32 dtype: object intakes In [356... ID Intake-Type Intake-Condition Intake-Age(days) **AnimalID** Intake-DateTime Out[356]: Breed Color Gender Name **0** A006100 A006100 2190 Spinone Italiano Yellow Scamp 2014-03-07 14:26:00 Public Assist Normal Male **Public Assist** A006100 A006100+ Spinone Italiano Yellow Scamp 2014-12-19 10:21:00 Normal 2555 Male A006100++ **2** A006100 Spinone Italiano Yellow Scamp 2017-12-07 14:07:00 Normal 3650 Male Stray A047759 A006101 Dachshund Tricolor Oreo 2014-04-02 15:55:00 Owner Surrender Normal 3650 Male 4 A134067 Shetland Sheepdog Public Assist A134067 Brown Male Bandit 2013-11-16 09:02:00 Injured 12190 A893570 87059 A893570 2023-11-23 12:17:00 730 Rottweiler Black Female <NA> Stray Normal 87060 A893573 A893573 **Border Terrier** NaN <NA> 2023-11-23 13:45:00 Stray Normal 730 Brown A893578 240 87061 A893578 American Staffordshire Terrier Yellow Brindle Male Tiger 2023-11-23 20:19:00 Stray Injured 87062 A893579 A893579 American Staffordshire Terrier Black Male <NA> 2023-11-23 21:00:00 Stray Injured 730 87063 A893585 A893585 German Shepherd Male Unknown 2023-11-23 20:19:00 30 Black Stray Injured 87064 rows × 10 columns # Read outcomes csv file outcomes = pd.read\_csv('processed\_data/dog\_outcomes\_unified.csv', dtype={ 'AnimalID': 'string', 'ID': 'string', 'Breed': 'string' 'Color': 'string', 'Gender': 'category', 'Name': 'string', 'Date-Of-Birth': 'string', Outcome-DateTime': 'string', 'Outcome-Type': 'string', 'Outcome-Subtype': 'string', 'Outcome-Age(days)': 'int' }, parse\_dates=['Date-Of-Birth', 'Outcome-DateTime']) outcomes.dtypes AnimalID string[python] Out[357]: string[python] ID Breed string[python] Color string[python] Gender category Name string[python] Date-Of-Birth datetime64[ns] Outcome-DateTime datetime64[ns] Outcome-Type string[python] Outcome-Subtype string[python] Outcome-Age(days) int32 dtype: object outcomes In [358.. ID Out[358]: AnimalID Breed Color Gender Name Date-Of-Birth Outcome-DateTime Outcome-Type Outcome-Subtype Outcome-Age(days) **0** A006100 A006100 2435 Spinone Italiano Yellow Male 2007-07-09 2014-03-08 17:10:00 Return to Owner <NA> Scamp A006100 A006100+ 2722 Spinone Italiano Yellow Male Scamp 2007-07-09 2014-12-20 16:35:00 Return to Owner <NA> A006100++ **2** A006100 Spinone Italiano Yellow Male 2007-07-09 2017-12-07 00:00:00 Return to Owner <NA> 3804 Scamp A047759 A047759 Dachshund Tricolor Male 2004-04-02 2014-04-07 15:12:00 Partner 3658 Oreo 4 A134067 <NA> A134067 Shetland Sheepdog Male 1997-10-16 2013-11-16 11:54:00 Return to Owner 5875 Brown Bandit A893431 86974 A893431 Chili 2015-11-21 2023-11-21 15:41:00 Return to Owner <NA> 2923 Chihuahua Tricolor Female 86975 A893432 A893432 Chihuahua Tan Female Coco 2015-11-21 2023-11-21 15:41:00 Return to Owner <NA> 2923 A893452 A893452 White <NA> 2557 86976 Maltese Sophie 2016-11-21 2023-11-22 11:26:00 Return to Owner Female 86977 A893529 A893529 Labrador Retriever <NA> 2023-09-22 2023-11-22 16:51:00 Transfer Partner 62 White Female A893585 62 86978 A893585 2023-09-24 2023-11-24 14:12:00 German Shepherd Black Male Unknown Transfer Partner 86979 rows × 11 columns Fill in missing values # Fill in missing value in Name and Outcome-Subtype with Unknwon intakes['Name'] = intakes['Name'].fillna('Unknown') outcomes['Name'] = outcomes['Name'].fillna('Unknown') outcomes['Outcome-Subtype'] = outcomes['Outcome-Subtype'].fillna('Unknown') # Rrop rows that still have missing value intakes = intakes.dropna().reset\_index(drop=True) outcomes = outcomes.dropna().reset\_index(drop=True) Remove Abnormal values # Remove negative age values intakes = intakes[intakes['Intake-Age(days)'] >= 0] outcomes = outcomes[outcomes['Outcome-Age(days)'] >= 0] intakes In [361... Intake-Type Intake-Condition Intake-Age(days) **AnimalID** ID **Breed** Color Gender Intake-DateTime Out[361]: Name **0** A006100 2014-03-07 14:26:00 A006100 Spinone Italiano Yellow Male Public Assist Normal 2190 A006100+ A006100 Spinone Italiano Yellow Scamp 2014-12-19 10:21:00 **Public Assist** Normal 2555 Male 2 A006100 A006100++ Spinone Italiano Yellow Male Scamp 2017-12-07 14:07:00 Stray Normal 3650 **3** A047759 A006101 Dachshund Oreo 2014-04-02 15:55:00 Owner Surrender Normal 3650 Tricolor Male 2013-11-16 00:02:00 A893565 Unknown 2023-11-23 08:31:00 86402 A893565 Chihuahua Buff Female Stray Injured 365 86403 A893570 A893570 Rottweiler 2023-11-23 12:17:00 Stray Normal 730 Black Female Unknown A893578 86404 A893578 American Staffordshire Terrier Yellow Brindle Male Tiger 2023-11-23 20:19:00 Stray Injured 240 86405 A893579 A893579 American Staffordshire Terrier Black Male Unknown 2023-11-23 21:00:00 Stray Injured 730 A893585 30 86406 A893585 German Shepherd Black Male Unknown 2023-11-23 20:19:00 Stray Injured 86396 rows × 10 columns outcomes In [362... AnimalID ID Color Gender Name Date-Of-Birth Outcome-DateTime Outcome-Type Outcome-Subtype Outcome-Age(days) Out[362]: Breed **0** A006100 A006100 Spinone Italiano Yellow Male 2007-07-09 2014-03-08 17:10:00 Unknown 2435 Scamp Return to Owner A006100 2007-07-09 2014-12-20 16:35:00 A006100+ Spinone Italiano Yellow Male Scamp Return to Owner Unknown 2722 2 A006100 A006100++ Spinone Italiano Yellow Male Scamp 2007-07-09 2017-12-07 00:00:00 Return to Owner Unknown 3804 A047759 A047759 Dachshund Tricolor Male Oreo 2004-04-02 2014-04-07 15:12:00 Transfer Partner 3658 4 A134067 A134067 Shetland Sheepdog 1997-10-16 2013-11-16 11:54:00 Return to Owner 5875 Brown Male Bandit Unknown 86301 A893431 A893431 Chihuahua Tricolor Female Chili 2015-11-21 2023-11-21 15:41:00 Return to Owner Unknown 2923 2015-11-21 2023-11-21 15:41:00 Return to Owner 86302 A893432 A893432 Chihuahua Tan Female Coco Unknown 2923 86303 A893452 A893452 Maltese White Female Sophie 2016-11-21 2023-11-22 11:26:00 Return to Owner Unknown 2557 Female Unknown 86304 A893529 A893529 Labrador Retriever White 2023-09-22 2023-11-22 16:51:00 Transfer Partner 62 86305 A893585 A893585 German Shepherd Black Male Unknown 2023-09-24 2023-11-24 14:12:00 Transfer Partner 62 86285 rows × 11 columns Add a new column of age category according to the outcomes age conditions = [ In [363.. (outcomes['Outcome-Age(days)'] <= 180),</pre> (outcomes['Outcome-Age(days)'] > 180) & (outcomes['Outcome-Age(days)'] <= 730), (outcomes['Outcome-Age(days)'] > 730) & (outcomes['Outcome-Age(days)'] <= 2555), (outcomes['Outcome-Age(days)'] > 2555) values = ['Baby', 'Young', 'Adult', 'Senior'] outcomes['Age'] = np.select(conditions, values) Add a new boolean column 'IsAdopted' according to 'Outcome-type' # Define adopted as adoption outcome-type In [364... adopted = ['Adoption'] outcomes['IsAdopted'] = np.where(outcomes['Outcome-Type'].isin(adopted), True, False) In [365... outcomes AnimalID ID Breed Color Gender Name Date-Of-Birth Outcome-DateTime Outcome-Type Outcome-Subtype Outcome-Age(days) Age IsAdopted Out[365]: A006100 A006100 Return to Owner Spinone Italiano 2007-07-09 2014-03-08 17:10:00 Unknown 2435 Adult False Yellow Male Scamp A006100 A006100+ 2007-07-09 2014-12-20 16:35:00 False Spinone Italiano Yellow Male Scamp Return to Owner Unknown 2722 Senior A006100++ **2** A006100 Spinone Italiano Yellow Male Scamp 2007-07-09 2017-12-07 00:00:00 Return to Owner Unknown 3804 Senior False A047759 A047759 Dachshund Tricolor Male Oreo 2004-04-02 2014-04-07 15:12:00 Transfer Partner 3658 Senior False A134067 A134067 1997-10-16 2013-11-16 11:54:00 Shetland Sheepdog Brown Male Bandit Return to Owner Unknown 5875 Senior False Chihuahua False A893432 A893432 2015-11-21 2023-11-21 15:41:00 Return to Owner 2923 Senior 86302 Chihuahua Tan Female Coco Unknown False White 86303 A893452 A893452 Maltese Sophie 2016-11-21 2023-11-22 11:26:00 Return to Owner Unknown 2557 Senior False Female 86304 A893529 A893529 Labrador Retriever 2023-09-22 2023-11-22 16:51:00 Partner 62 Baby False White Female Unknown Transfer A893585 Baby 86305 A893585 2023-09-24 2023-11-24 14:12:00 62 False German Shepherd Black Male Unknown Transfer Partner 86285 rows × 13 columns Merge intakes and outcomes dataframes by ID intakes\_outcomes = pd.merge(intakes, outcomes, how="inner", on=["AnimalID", "ID", "Breed", "Color", "Gender", "Name"], validate="1:1") In [366... intakes\_outcomes['Outcome-Age(days)'] = intakes\_outcomes['Outcome-Age(days)'].astype('Int64') intakes\_outcomes Out[366]: Date-Intake-Intake-Intake-Outcome-Outcome-Outcome-Outcome-Intake-Age IsAdopted **AnimalID** ID Color Gender Name Of-**Breed DateTime** Condition Subtype Type Age(days) **DateTime** Type Age(days) Birth 2007-2014-03-07 Public 2014-03-08 Return to Spinone A006100 0 A006100 Yellow Male Scamp Normal 2190 Unknown 2435 Adult False Italiano 14:26:00 Assist 07-09 17:10:00 Owner **Public** 2014-12-19 2007-2014-12-20 Spinone Return to **1** A006100 A006100+ Yellow Male Scamp Normal 2555 Unknown 2722 Senior False Italiano 10:21:00 07-09 16:35:00 Assist Owner 2017-12-07 2017-12-07 Return to Spinone 2007-Stray 2 A006100 A006100++ Yellow Male Scamp Normal 3650 Unknown 3804 Senior False Italiano 14:07:00 07-09 00:00:00 Owner Public 2013-11-16 1997-2013-11-16 Shetland Return to **3** A134067 A134067 Brown Male Bandit Injured 12190 Unknown 5875 Senior False 09:02:00 10-16 11:54:00 Sheepdog Assist Owner 1998-2013-11-17 2013-11-16 Return to Labrador Stray **4** A141142 A141142 Black Female Bettie Aged 11825 Unknown 5648 Senior False Retriever 14:46:00 06-01 11:40:00 Owner 2023-11-21 Public 2015-2023-11-21 Return to 2923 Senior 85791 A893431 A893431 Chihuahua Tricolor Female Chili Normal 2920 Unknown False 11:21:00 11-21 15:41:00 Assist Owne 2023-11-21 2023-11-21 Public 2015-Return to 85792 A893432 A893432 Chihuahua Tan Female Coco Normal 2920 Unknown 2923 Senior False 11:21:00 11-21 15:41:00 Assist Owne 2023-11-21 Public 2023-11-22 2016-Return to 85793 A893452 A893452 Maltese White Female Sophie Normal 2555 Unknown 2557 Senior False 11:26:00 13:38:00 11-21 Assist Owner 2023-2023-11-22 Labrador 2023-11-22 Owner A893529 85794 A893529 White Female Unknown Normal 30 Transfer Partner 62 Baby False 09-22 Retriever 14:26:00 16:51:00 Surrender 2023-11-23 2023-11-24 German 2023-A893585 30 85795 A893585 Black Male Unknown Stray Injured Transfer Partner 62 Baby False 09-24 Shepherd 20:19:00 14:12:00 85796 rows × 17 columns Read dog breed data breeds = pd.read\_csv('processed\_data/breed\_data.csv', dtype={ In [367... 'Breed': 'string', 'Category': 'category', 'Intelligence-Ranking': 'Float64', 'Intelligence-Category': 'category', 'Size-Category': 'category', 'Longevity': 'Float64', 'Total-Cost(\$)': 'Float64' 'Purchase-Cost(\$)': 'Float64', 'Food-Cost(\$)': 'Float64' breeds.dtypes Breed string[python] Out[367]: category Category Intelligence-Ranking Float64 Intelligence-Category category Size-Category category Longevity Float64 IOTAL-COST(\$) FIOat64 Purchase-Cost(\$) Float64 Float64 Food-Cost(\$) dtype: object breeds In [368... Category Intelligence-Ranking Intelligence-Category Size-Category Longevity Total-Cost(\$) Purchase-Cost(\$) Food-Cost(\$) Out[368]: Breed 15835.00 0 Affenpinscher Toy 37.00 Above-Average Small 11.42 510.00 3180.00 1 Afghan Hound Hound 80.00 Lowest Large 11.92 20818.00 890.00 7260.00 2 Above-Average Airedale Terrier Terrier 29.00 Medium 11.45 <NA> 733.00 <NA> 3 Akita Working 54.00 Large 10.16 18217.00 1202.00 6188.00 Average 4 Alaskan Malamute Working 50.00 Average Large 10.67 19069.00 1210.00 6499.00 167 Whippet Hound 51.00 Average Medium 12.87 18160.00 915.00 3584.00 168 Wire Fox Terrier 51.00 Small 13.17 <NA> 668.00 <NA> Terrier Average 169 Wirehaired Pointing Griffon Sporting 46.00 Average Medium 8.80 <NA> 755.00 <NA>170 Xoloitzcuintli Non-Sporting <NA> Medium <NA> <NA> 717.00 <NA> NaN 171 27.00 Yorkshire Terrier Above-Average Small 12.60 17944.00 1057.00 3508.00 172 rows × 9 columns Fill in missing values In [369... # Fill in missing values in Intelligence-Ranking and Longevity with the average value of the same category breeds["Intelligence-Ranking"].fillna( breeds.groupby("Category")["Intelligence-Ranking"].transform("mean"),inplace=True breeds["Intelligence-Ranking"].fillna( breeds["Intelligence-Ranking"].mean(),inplace=True breeds["Longevity"].fillna( breeds.groupby("Category")["Longevity"].transform("mean"),inplace=True breeds["Longevity"].fillna( breeds["Longevity"].mean(),inplace=True # Fill in numeric missing values in Cost with average value of the same category & size combination breeds["Total-Cost(\$)"].fillna( breeds.groupby(["Category", "Size-Category"])["Total-Cost(\$)"].transform("mean"),inplace=True breeds["Total-Cost(\$)"].fillna( breeds["Total-Cost(\$)"].mean(),inplace=True breeds["Food-Cost(\$)"].fillna( breeds.groupby(["Category", "Size-Category"])["Food-Cost(\$)"].transform("mean"),inplace=True breeds["Food-Cost(\$)"].fillna( breeds["Food-Cost(\$)"].mean(),inplace=True breeds["Purchase-Cost(\$)"].fillna( breeds.groupby(["Category", "Longevity", "Size-Category"])["Purchase-Cost(\$)"].transform("mean"),inplace=True breeds["Purchase-Cost(\$)"].fillna( breeds["Purchase-Cost(\$)"].mean(),inplace=True breeds Category Intelligence-Ranking Intelligence-Category Size-Category Longevity Total-Cost(\$) Purchase-Cost(\$) Food-Cost(\$) Out[370]: **Breed** 0 Affenpinscher Toy 37.00 Small 11.42 15835.00 510.00 3180.00 Above-Average 1 80.00 11.92 20818.00 890.00 7260.00 Afghan Hound Hound Lowest Large 2 29.00 16605.67 733.00 4224.00 Airedale Terrier Terrier Above-Average Medium 11.45 3 54.00 10.16 18217.00 1202.00 6188.00 Akita Working Average Large Alaskan Malamute 50.00 1210.00 6499.00 4 Working Average Large 10.67 19069.00 ••• 51.00 18160.00 915.00 3584.00 167 Whippet Hound Average Medium 12.87 51.00 13.17 18078.44 668.00 3809.00 168 Wire Fox Terrier Terrier Average Small 169 Wirehaired Pointing Griffon 46.00 Medium 8.80 16668.89 755.00 4219.45 Sporting Average 717.00 170 Xoloitzcuintli Non-Sporting 46.71 NaN Medium 10.98 15294.33 3790.29 171 Yorkshire Terrier 27.00 Above-Average Small 12.60 17944.00 1057.00 3508.00 Toy 172 rows × 9 columns # Cast object numbers to float In [371... pd.options.display.float\_format = "{:.2f}".format breeds["Longevity"] = breeds["Longevity"].astype('float') # Cast object numbers to int breeds["Intelligence-Ranking"] = breeds["Intelligence-Ranking"].astype('int') breeds["Total-Cost(\$)"] = breeds["Total-Cost(\$)"].astype('int') breeds["Purchase-Cost(\$)"] = breeds["Purchase-Cost(\$)"].astype('int') breeds["Food-Cost(\$)"] = breeds["Food-Cost(\$)"].astype('int') breeds.dtypes Breed string[python] Out[371]: Category category Intelligence-Ranking int32 Intelligence-Category category Size-Category category Longevity float64 Total-Cost(\$) int32 Purchase-Cost(\$) int32 Food-Cost(\$) int32 dtype: object # Fill in missing values in Intelligence-Category according to Intelligence-Ranking In [372... intel\_conditions = [ (breeds['Intelligence-Ranking'] <= 10),</pre> (breeds['Intelligence-Ranking'] > 10) & (breeds['Intelligence-Ranking'] <= 26),</pre> (breeds['Intelligence-Ranking'] > 26) & (breeds['Intelligence-Ranking'] <= 39),</pre> (breeds['Intelligence-Ranking'] > 39) & (breeds['Intelligence-Ranking'] <= 54),</pre> (breeds['Intelligence-Ranking'] > 54) & (breeds['Intelligence-Ranking'] <= 69),</pre> (breeds['Intelligence-Ranking'] > 69) intel\_values = ['Brightest', 'Excellent', 'Above-Average', 'Average', 'Fair', 'Low'] breeds['Intelligence-Category'] = np.select(intel\_conditions, intel\_values) breeds Out[372]: Breed Category Intelligence-Ranking Intelligence-Category Size-Category Longevity Total-Cost(\$) Purchase-Cost(\$) Food-Cost(\$) 11.42 0 Affenpinscher 37 15835 510 3180 Toy Above-Average Small 1 80 11.92 20818 890 7260 Afghan Hound Hound Low Large 2 Airedale Terrier Terrier 29 Above-Average Medium 11.45 16605 733 4224 3 54 10.16 18217 1202 6188 Akita Working Average Large Alaskan Malamute 50 19069 1210 4 Working 10.67 6499 Average Large 51 12.87 18160 167 Whippet Hound Average Medium 915 3584 168 Wire Fox Terrier Terrier 51 Average Small 13.17 18078 668 3809 46 4219 **169** Wirehaired Pointing Griffon Medium 8.80 16668 755 Sporting Average 170 Medium 10.98 15294 717 3790 Xoloitzcuintli Non-Sporting Average 172 rows × 9 columns Use Agglomerative Clustering to add a new column 'Cost-Category' based on Total-Cost, Purchase-Cost, Food-Cost In [373... **from** sklearn.cluster **import** AgglomerativeClustering import scipy.cluster.hierarchy as shc cost\_df = breeds[['Total-Cost(\$)', 'Purchase-Cost(\$)', 'Food-Cost(\$)']] cost\_df Total-Cost(\$) Purchase-Cost(\$) Food-Cost(\$) Out[374]: 15835 510 3180 0 1 20818 890 7260 2 16605 733 4224 18217 1202 3 6188 19069 1210 4 6499 ••• 167 18160 915 3584 18078 168 668 3809 169 16668 755 4219 15294 170 717 3790 171 17944 1057 3508 172 rows × 3 columns dend\_ward = shc.dendrogram(shc.linkage(cost\_df, method='ward')) 30000 25000 20000 15000 10000 5000 In [376... # Clustering - ward linkage # Choose number of clusters as 4 according to dendrogram clustering\_ward = AgglomerativeClustering(linkage='ward', n\_clusters=4) clustering\_ward.fit(cost\_df) breeds['Cost-Category'] = clustering\_ward.labels\_  $current_labels = [0, 1, 2, 3]$ desired\_labels = ['Saver', 'Budget', 'Mid-Price', 'High-End'] map\_dict = dict(zip(current\_labels, desired\_labels)) breeds['Cost-Category'] = breeds['Cost-Category'].map(map\_dict) breeds Out[376]: Category Intelligence-Ranking Intelligence-Category Size-Category Longevity Total-Cost(\$) Purchase-Cost(\$) Food-Cost(\$) Cost-Category Breed 0 Affenpinscher Toy 37 Above-Average Small 11.42 15835 510 3180 Saver 20818 High-End 1 Afghan Hound Hound 80 Low Large 11.92 890 7260 2 Airedale Terrier Terrier 29 Above-Average Medium 11.45 16605 733 4224 Saver 3 54 1202 6188 Working 10.16 18217 Akita Average Large Budget 4 Alaskan Malamute 50 19069 1210 6499 Working Average Large 10.67 Budget 51 18160 915 167 Whippet Hound Average Medium 12.87 3584 Budget 168 Wire Fox Terrier 51 13.17 18078 668 3809 Terrier Average Small Budget 169 Wirehaired Pointing Griffon 46 16668 4219 Sporting Average Medium 8.80 755 Saver 170 46 15294 3790 Saver Medium 10.98 717 Xoloitzcuintli Non-Sporting Average 171 27 Yorkshire Terrier Toy Above-Average Small 12.60 17944 1057 3508 Budget 172 rows × 10 columns In [377...] fig = plt.figure(figsize = (10, 7)) ax = plt.axes(projection ="3d") scatter = ax.scatter3D(cost\_df['Purchase-Cost(\$)'], cost\_df['Food-Cost(\$)'], cost\_df['Total-Cost(\$)'], c = clustering\_ward.labels\_) ax.set(xlabel='Purchase-Cost(\$)', ylabel='Food-Cost(\$)', zlabel='Total-Cost(\$)')  $_{-}$  = ax.legend( scatter.legend\_elements()[0], desired\_labels, loc="lower right", title="Classes" plt.show() 22000 -20000\_ 180000 -16000£ 14000 †12000 10000 9000 8000 Classes 1500 2000 2500 Saver 1000 Purchase-Cost(\$)Budget 4000 Mid-Price 3000 3000 High-End 3500 Join breed data to intakes & outcomes data data = pd.merge(intakes\_outcomes, breeds, how="left", on=["Breed"]) Intelligence- Intelligence-Intake-Intake-Intake-Total- P Out[378]: Intake-Size-... IsAdopted Category **AnimalID** ID **Breed** Color Gender Name Longevity DateTime Type Condition Age(days) Ranking **Category Category** Cost(\$) 2014-03-**Public** Spinone Above-A006100 A006100 2190 ... Yellow Male 07 Normal False Sporting 27 Large 9.00 18062 Scamp Italiano **Assist** Average 14:26:00 2014-12-**Public** Above-Spinone A006100+ **1** A006100 Yellow 27 9.00 18062 Male Scamp 19 Normal 2555 ... False Sporting Large Italiano Assist Average 10:21:00 2017-12-Spinone Above-2 A006100 A006100++ 27 18062 Yellow Male Scamp 07 Stray Normal 3650 ... False Sporting Large 9.00 Italiano Average 14:07:00 2013-11-Shetland Public **3** A134067 A134067 12190 ... 12.53 17469 Brown Male Bandit 16 Injured False Herding 6 Brightest Small Sheepdog Assist 09:02:00 2013-11-Labrador **4** A141142 A141142 11825 ... 7 **Brightest** 12.04 18422 Black Female Bettie 16 Stray Aged False Sporting Medium Retriever 14:46:00 2023-11-**Public** 2920 ... False 67 16.50 22640 **85791** A893431 A893431 Chihuahua Tricolor Female Chili 21 Normal Toy Fair Small **Assist** 11:21:00 2023-11-**Public** A893432 Chihuahua A893432 67 16.50 22640 85792 Tan Female Coco 21 Normal 2920 ... False Toy Fair Small **Assist** 11:21:00 2023-11-**Public** A893452 A893452 59 16073 85793 Maltese White Female Sophie 21 Normal 2555 ... False Toy Fair Small 12.25 **Assist** 13:38:00 2023-11-Labrador Owner 85794 A893529 A893529 7 Brightest 12.04 18422 White Female Unknown 22 Normal 30 False Sporting Medium Surrender Retriever 14:26:00 2023-11-German Herding 85795 A893585 A893585 Black Male Unknown 23 Stray Injured 30 ... False 3 Brightest Large 9.73 15091 Shepherd 20:19:00 85796 rows × 26 columns Read active dogs data from petfinder In [379... petfinder = pd.read\_csv('processed\_data/petfinder\_unified.csv', dtype={ 'Petfinder-ID': 'string', 'Name': 'string', 'Size': 'category' 'Gender': 'category', 'Age': 'category', 'Color': 'string', 'Breed': 'string' petfinder.dtypes Petfinder-ID string[python] Out[379] Name string[python] Size category Gender category Age category Color string[python] Breed string[python] dtype: object In [380... # Drop rows that have a missing value petfinder = petfinder.dropna().reset\_index(drop=True) petfinder In [381... Petfinder-ID Name Gende Color Breed Large 65653819 Eloise Female Adult Gray **Bull Terrier** 68926034 Black Labrador Retriever Penny Medium Female Baby 2 69313950 Saint Bernard Pirate Extra Large Male Baby Tricolor 69340682 3 Fiona Chihuahua Small Female Senior Apricot 69402276 Pembroke Welsh Corgi Rory Medium Female Adult Red ... 7334 69816070 **Bubba and Cadence** Small Male Senior Tricolor **Border Terrier** 7335 69816083 Bubba Large Male Young Black Parson Russell Terrier 69816084 NORA - 3 mo. old gentle guiet smart curious baby Large Labrador Retriever 7336 Female Baby Yellow 7337 69816100 Blaze Medium Male Baby Black American Staffordshire Terrier 7338 69816108 Molly Medium Female Baby Red Australian Cattle Dog 7339 rows × 7 columns Join breed data to petfinder dog data petfinder = pd.merge(petfinder, breeds, how="left", on=["Breed"]) In [382... petfinder Petfinder-Intelligence-Intelligence-Purchase-Out[382]: Size-Total-Food-Cost-Name Size Gender Color **Breed Category** Longevity Age Ranking Category Category Cost(\$) Cost(\$) Cost(\$) Category 65653819 66 Fair 10.21 16051 1085 4087 0 Eloise Large Female Adult Gray **Bull Terrier** Terrier Medium Saver Labrador 68926034 Penny Medium Female Baby Black Sporting 7 Brightest Medium 12.04 18422 810 4819 Budget Retriever Extra Saint Bernard 65 2 69313950 Pirate Male Baby Tricolor Working Fair Large 7.78 17336 875 8124 Saver Large 69340682 Chihuahua 67 16.50 22640 588 4594 High-End Fiona Small Female Senior Apricot Toy Fair Small Pembroke 69402276 Herding 11 Excellent 19625 587 6026 Rory Medium Female Adult Red Small 12.25 Budget Welsh Corgi Bubba and Above-7334 69816070 Small Male Senior Tricolor **Border Terrier** Terrier 30 Small 14.00 19575 833 3898 Budget Cadence Average Parson Russell 7335 69816083 Bubba Male Young Black Terrier 44 Average Small 11.48 18078 528 3809 Budget Large Terrier NORA - 3 mo. old Labrador 7 18422 4819 69816084 Sporting Brightest 12.04 810 7336 gentle quiet smart Large Female Baby Yellow Medium Budget Retriever curious baby American Above-69816100 34 11.48 16605 1043 4224 7337 Blaze Medium Male Baby Black Staffordshire Terrier Medium Saver Average Terrier Australian Cattle 69816108 7338 Molly Medium Female Baby Red Herding 10 Brightest Medium 11.67 17607 530 4671 Saver Dog 7339 rows × 16 columns Write the processed data to csv file data.to\_csv('data.csv', index=False) In [383... petfinder.to\_csv('petfinder.csv', index=False) features = data[['Breed', 'Color', 'Gender', 'Age', 'Intake-Type', 'Category', 'Intelligence-Category', 'Size-Category', 'Cost-Category', 'IsAdopted']] features.to\_csv('features.csv', index=False, header=False)