Predict Health Outcomes of Horses

Reading the dataset into a pandas DataFrame

In [57]:
```
train = pd.read_csv('/kaggle/input/playground-series-s3e22/train.csv')
test = pd.read_csv('/kaggle/input/playground-series-s3e22/test.csv')
sub = pd.read_csv('/kaggle/input/playground-series-s3e22/sample_submission.csv')
```

Convert the categorical data type into numerical data type on the target feature "outcome"

In [58]:
```
train["outcome"] = train["outcome"].map({"died": 0, "euthanized": 1, "lived": 2})
```

In [59]:
```
train['outcome']
```

Out[59]:
```
0       0
1       1
2       2
3       2
4       2
       ..
1230    2
1231    0
1232    2
1233    2
1234    2
Name: outcome, Length: 1235, dtype: int64
```

Descriptive data analysis on the train dataset

In [60]:
```
train.head()
```

Out[60]:

	id	surgery	age	hospital_number	rectal_temp	pulse	respiratory_rate	temp_of_extremities	peri
0	0	yes	adult	530001	38.1	132.0	24.0	cool	
1	1	yes	adult	533836	37.5	88.0	12.0	cool	
2	2	yes	adult	529812	38.3	120.0	28.0	cool	
3	3	yes	adult	5262541	37.1	72.0	30.0	cold	
4	4	no	adult	5299629	38.0	52.0	48.0	normal	

5 rows × 29 columns

In [61]:
```
test.head()
```

Out[61]:

	id	surgery	age	hospital_number	rectal_temp	pulse	respiratory_rate	temp_of_extremities	p
0	1235	no	adult	534053	38.6	40.0	20.0	normal	
1	1236	yes	adult	528469	38.2	112.0	48.0	cool	
2	1237	yes	adult	528178	37.7	66.0	12.0	cool	
3	1238	no	adult	534784	37.1	88.0	20.0	cool	
4	1239	yes	adult	529840	38.3	50.0	12.0	NaN	

5 rows × 28 columns

In [62]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1235 entries, 0 to 1234
Data columns (total 29 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     1235 non-null   int64
 1   surgery                1235 non-null   object
 2   age                    1235 non-null   object
 3   hospital_number        1235 non-null   int64
 4   rectal_temp            1235 non-null   float64
 5   pulse                  1235 non-null   float64
 6   respiratory_rate       1235 non-null   float64
 7   temp_of_extremities    1196 non-null   object
 8   peripheral_pulse       1175 non-null   object
 9   mucous_membrane        1214 non-null   object
 10  capillary_refill_time  1229 non-null   object
 11  pain                   1191 non-null   object
 12  peristalsis            1215 non-null   object
 13  abdominal_distention   1212 non-null   object
 14  nasogastric_tube       1155 non-null   object
 15  nasogastric_reflux     1214 non-null   object
 16  nasogastric_reflux_ph  1235 non-null   float64
 17  rectal_exam_feces      1045 non-null   object
 18  abdomen                1022 non-null   object
 19  packed_cell_volume     1235 non-null   float64
 20  total_protein          1235 non-null   float64
 21  abdomo_appearance      1187 non-null   object
 22  abdomo_protein         1235 non-null   float64
 23  surgical_lesion        1235 non-null   object
 24  lesion_1               1235 non-null   int64
 25  lesion_2               1235 non-null   int64
 26  lesion_3               1235 non-null   int64
 27  cp_data                1235 non-null   object
 28  outcome                1235 non-null   int64
dtypes: float64(7), int64(6), object(16)
memory usage: 279.9+ KB
```

In [63]:
```
train.isnull().sum()
```

Out[63]:
```
id                        0
surgery                   0
age                       0
hospital_number           0
rectal_temp               0
pulse                     0
respiratory_rate          0
temp_of_extremities      39
peripheral_pulse         60
mucous_membrane          21
capillary_refill_time     6
pain                     44
peristalsis              20
abdominal_distention     23
nasogastric_tube         80
nasogastric_reflux       21
nasogastric_reflux_ph     0
rectal_exam_feces       190
abdomen                 213
packed_cell_volume        0
total_protein             0
abdomo_appearance        48
abdomo_protein            0
surgical_lesion           0
lesion_1                  0
lesion_2                  0
lesion_3                  0
cp_data                   0
outcome                   0
dtype: int64
```

In [64]:
```
train.duplicated().sum()
```

Out[64]:
```
0
```

In [65]:
```
print('train shape: ', train.shape)
print('test shape: ', test.shape)
```

```
train shape:  (1235, 29)
test shape:  (824, 28)
```

In [66]:
```
train.dtypes
```

Out[66]:
```
id                      int64
surgery                object
age                    object
hospital_number         int64
rectal_temp           float64
pulse                 float64
respiratory_rate      float64
temp_of_extremities    object
peripheral_pulse       object
mucous_membrane        object
capillary_refill_time  object
pain                   object
peristalsis            object
abdominal_distention   object
nasogastric_tube       object
nasogastric_reflux     object
nasogastric_reflux_ph  float64
rectal_exam_feces      object
abdomen                object
packed_cell_volume    float64
total_protein         float64
abdomo_appearance      object
abdomo_protein        float64
surgical_lesion        object
lesion_1                int64
lesion_2                int64
lesion_3                int64
cp_data                object
outcome                 int64
dtype: object
```

Unique Values in Given Columns

In [67]:
```
columns = ['surgery',
 'age',
 'hospital_number',
 'rectal_temp',
 'pulse',
 'respiratory_rate',
 'temp_of_extremities',
 'peripheral_pulse',
 'mucous_membrane',
 'capillary_refill_time',
 'pain',
 'peristalsis',
 'abdominal_distention',
 'nasogastric_tube',
 'nasogastric_reflux',
 'nasogastric_reflux_ph',
 'rectal_exam_feces',
 'abdomen',
 'packed_cell_volume',
 'total_protein',
 'abdomo_appearance',
 'abdomo_protein',
 'surgical_lesion',
 'lesion_1',
 'lesion_2',
 'lesion_3',
 'cp_data',
```

```
 'outcome']
for col in columns:
    print(col, train[col].unique())
```

```
surgery ['yes' 'no']
age ['adult' 'young']
hospital_number [ 530001  533836  529812 5262541 5299629  529642  534787  529461  528
742
  529640  528682  530028  528548  528134  528305  534885 5290482 5279822
  533692  535208  528523  529893  534145  530233  529399  530354  528503
  529796  527916  530360  528298  533871  529388  527563  534163  529827
  535196  535176  529045  527518  527463  529172  528996  533887  528904
  535407  533902  523190  534073  534135 5290409  529160  534917  534784
 5299253  534004  534115  529667 5297159  529427  527677  530612  535415
  530561  530242  530002 5289419  529498  529126 5291409 5287179  530526
 5290759  532110  534293  534280  528214  527933 5283431  528743  529766
  529304  530401  527702  529849  534157  534998  529340  530276  527927
  534886  527365  528641  528461  528469  529607  533942  529272  528183
  533696  528247  535043  530034  534925  530402  535292  529493  528355
  534197  530239  529777  530478  534069  533928  528570  528800  528668
  530693  528179  528151  530254  528890  534644  530255  529663  535031
  534403  528804  528872  528702  535166  535330  528729  529888  529475
 5294539  533750  530334  535130  530439  534624  529628  530366  526639
 5301219  529703  528178  527734  529840  534753  534963 5291329  533885
  529518 5297379  534519  521399  529685  529183  530301  527544  527709
 5294369  528248  527758  535085  534933  529428  533847  530251  533738
  527706  527698  529865  529373  534579  528620  529135  527526  529424
  528630  528006  530310  526802  535029  532985  534183  528299  534092
  530544  534788  530381  534324  530384  527957  533723  535163  522979
  528931  534756  535158  530670 5290481  527883  534053  530170 5291719
  534556  529736  533736  534626  534478 5290402  534938 5288249  535240
  530431 5292489  529386  535381  535137  533968  532349  535338  528019
  529483  529821  528812  535054  529729 5299603  535392  533886  534817
  529528  534130  529764  530101  533983  528113  527524 5279442 5275212
  530297 5287279 5305129  530624  535246  527829  528653 5278331 5282839
  534857  534491  530157]
rectal_temp [38.1 37.5 38.3 37.1 38.  39.2 37.4 39.3 37.8 38.8 38.7 37.6 39.1 39.4
 40.3 38.5 37.2 38.2 38.4 37.7 36.1 38.6 37.9 38.9 36.5 37.3 39.5 36.8
 36.  35.4 39.  39.6 36.4 39.8 40.8 37.  39.7 36.6 36.7 36.9 39.9 36.2
 40. ]
pulse [132.  88. 120.  72.  52.  56.  36. 114.  48. 129.  84. 164.  66. 124.
  44.  60.  96.  70.  92.  64.  80.  68. 140. 130. 108.  30. 100. 104.
  76. 146.  54.  50.  42.  40. 150.  45. 136.  86. 112.  90.  78.  49.
 128. 184.  98.  75.  82. 160.  46. 110.]
respiratory_rate [24. 12. 28. 30. 48. 32. 16. 36. 88. 20. 40. 96. 68. 18. 14. 35. 51.
13.
 21. 84. 50.  9. 80. 44.  8. 42. 90. 60. 58. 22. 25. 10. 26. 70. 66. 52.
 23.]
temp_of_extremities ['cool' 'cold' 'normal' 'warm' nan]
peripheral_pulse ['reduced' 'normal' nan 'absent' 'increased']
mucous_membrane ['dark_cyanotic' 'pale_cyanotic' 'pale_pink' 'normal_pink' 'bright_pi
nk'
 'bright_red' nan]
capillary_refill_time ['more_3_sec' 'less_3_sec' nan '3']
pain ['depressed' 'mild_pain' 'extreme_pain' 'alert' 'severe_pain' nan 'slight']
peristalsis ['absent' 'hypomotile' 'normal' 'hypermotile' nan 'distend_small']
abdominal_distention ['slight' 'moderate' 'none' 'severe' nan]
nasogastric_tube ['slight' 'none' 'significant' nan]
nasogastric_reflux ['less_1_liter' 'more_1_liter' 'none' nan 'slight']
nasogastric_reflux_ph [6.5 2.  3.5 7.  5.  4.5 4.4 5.3 6.  3.  7.5 5.5 5.7 1.  4.  5.
4 7.2 6.2
 5.2 4.3 1.5 6.8 4.2 3.4 2.6 6.6]
rectal_exam_feces ['decreased' 'absent' nan 'normal' 'increased' 'serosanguious']
abdomen ['distend_small' 'distend_large' 'normal' 'firm' nan 'other']
```

```
packed_cell_volume [57.  33.  37.  53.  47.  49.  43.  40.  46.  65.  44.  60.  39.
52.
 72.  35.  66.  34.  45.  59.  48.  73.  31.5 54.  30.  55.  64.  50.
 69.  36.  42.  68.  41.  75.  38.  51.  37.5 23.  63.  26.  28.  71.
 58.  56.  32.  31.  74.  67.  70. ]
total_protein [ 8.5 64.   6.4  7.   7.3  8.  75.   7.6  7.8  4.9  7.5  6.7  7.4 13.
  6.5  6.8  8.4 53.  65.   6.1 66.   4.6  5.3  8.6 67.   6.   7.2  8.1
 70.  85.  60.  82.   6.6  4.5  6.2 81.  62.   8.2  7.7 74.   5.9  3.9
 86.   4.  77.   5.5  5.7 11.   9.1 46.  69.   8.9 57.  51.  56.   8.7
 10.2 80.  61.  54.   9.  71.   8.3  6.3 55.  89.   7.9 68.   3.7  6.9
  8.8 11.2 50.   3.5  5.8  4.3  5.  84.   4.7 63.   7.1 72.  58. ]
abdomo_appearance ['serosanguious' 'cloudy' 'clear' nan]
abdomo_protein [ 3.4  2.   3.9  2.6  2.8  1.   4.5  2.9  3.6  6.6  3.3  4.1  5.3  3.7
  1.3  1.4  2.5  5.   0.1  3.2  1.5  4.4  3.   2.3  7.4  4.3  4.7  4.8
  6.   6.5  2.2  7.   1.6  5.2  2.7  5.4  3.5  2.1  4.6  7.6 10.1  1.7
  8.   8.1  7.2  6.3  4.2  4.9  7.5 10.   2.4  6.2  9.1  4. ]
surgical_lesion ['yes' 'no']
lesion_1 [ 2209  2208  5124     0  3111  2207  3209  3205  2124  2206 31110  2205
  7111  3207  4206  2113  3113  2112  4205  8300  1400  5400  7209  3115
 11124  4207  9400   300  2111  3300  3112   400  2300  2322  3133  4300
  3025  8400  1111  5206 11300  4124 12208  6112  7400  5000  5205  2202
  3124  5111  6209 11400  6111  2305 21110  1124 41110]
lesion_2 [   0 1400 3111 3112]
lesion_3 [   0 2209]
cp_data ['no' 'yes']
outcome [0 1 2]
```

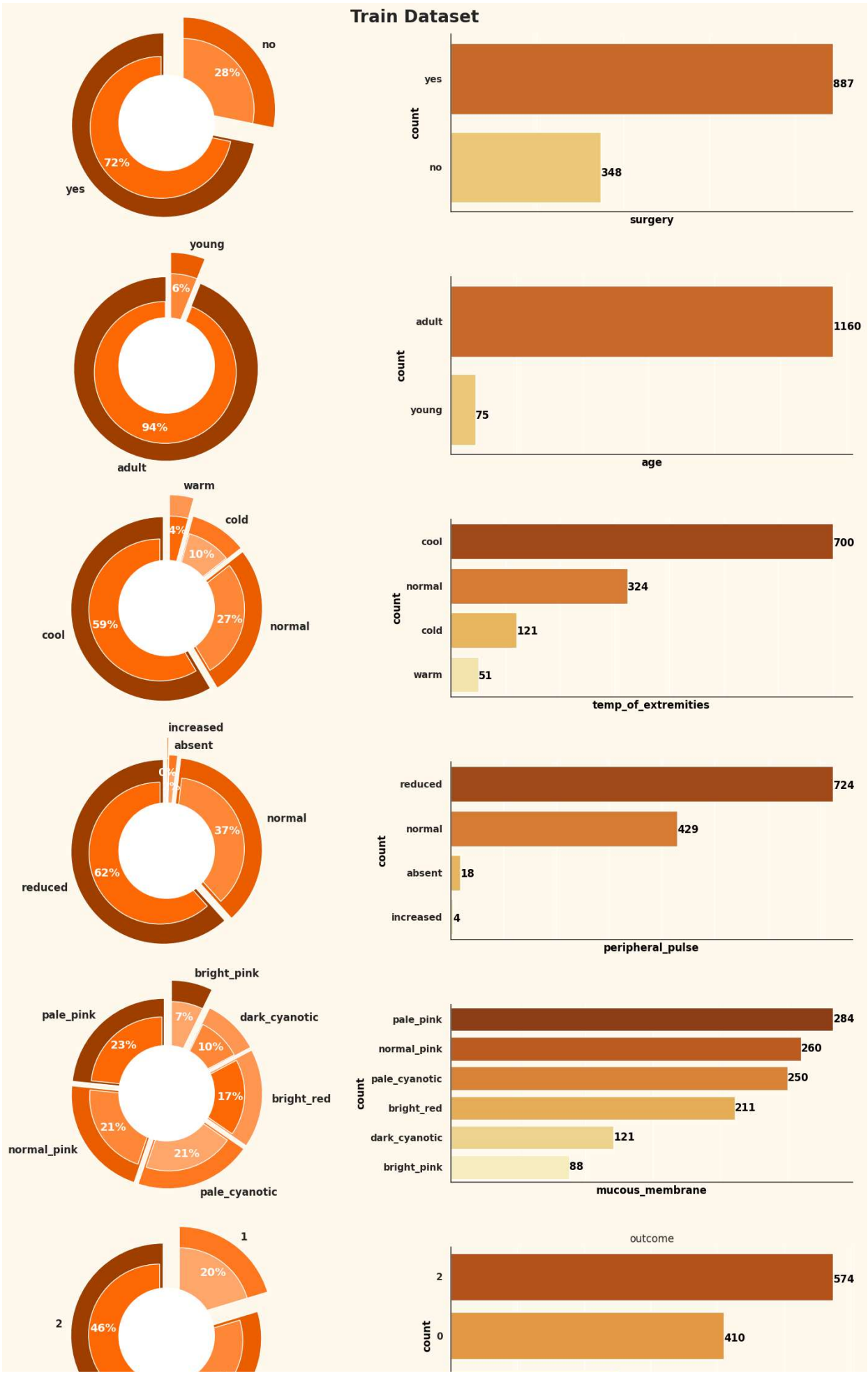
In [68]:
```
train.describe()
```

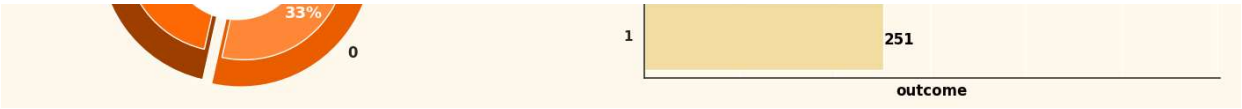
Out[68]:

	id	hospital_number	rectal_temp	pulse	respiratory_rate	nasogastric_reflux_ph
count	1235.0000	1.235000e+03	1235.000000	1235.000000	1235.000000	1235.000000
mean	617.0000	9.545004e+05	38.202186	79.574089	30.054251	4.382591
std	356.6581	1.356403e+06	0.788668	29.108638	16.452066	1.937357
min	0.0000	5.213990e+05	35.400000	30.000000	8.000000	1.000000
25%	308.5000	5.288000e+05	37.800000	53.000000	18.000000	2.000000
50%	617.0000	5.297770e+05	38.200000	76.000000	28.000000	4.500000
75%	925.5000	5.341450e+05	38.600000	100.000000	36.000000	6.000000
max	1234.0000	5.305129e+06	40.800000	184.000000	96.000000	7.500000

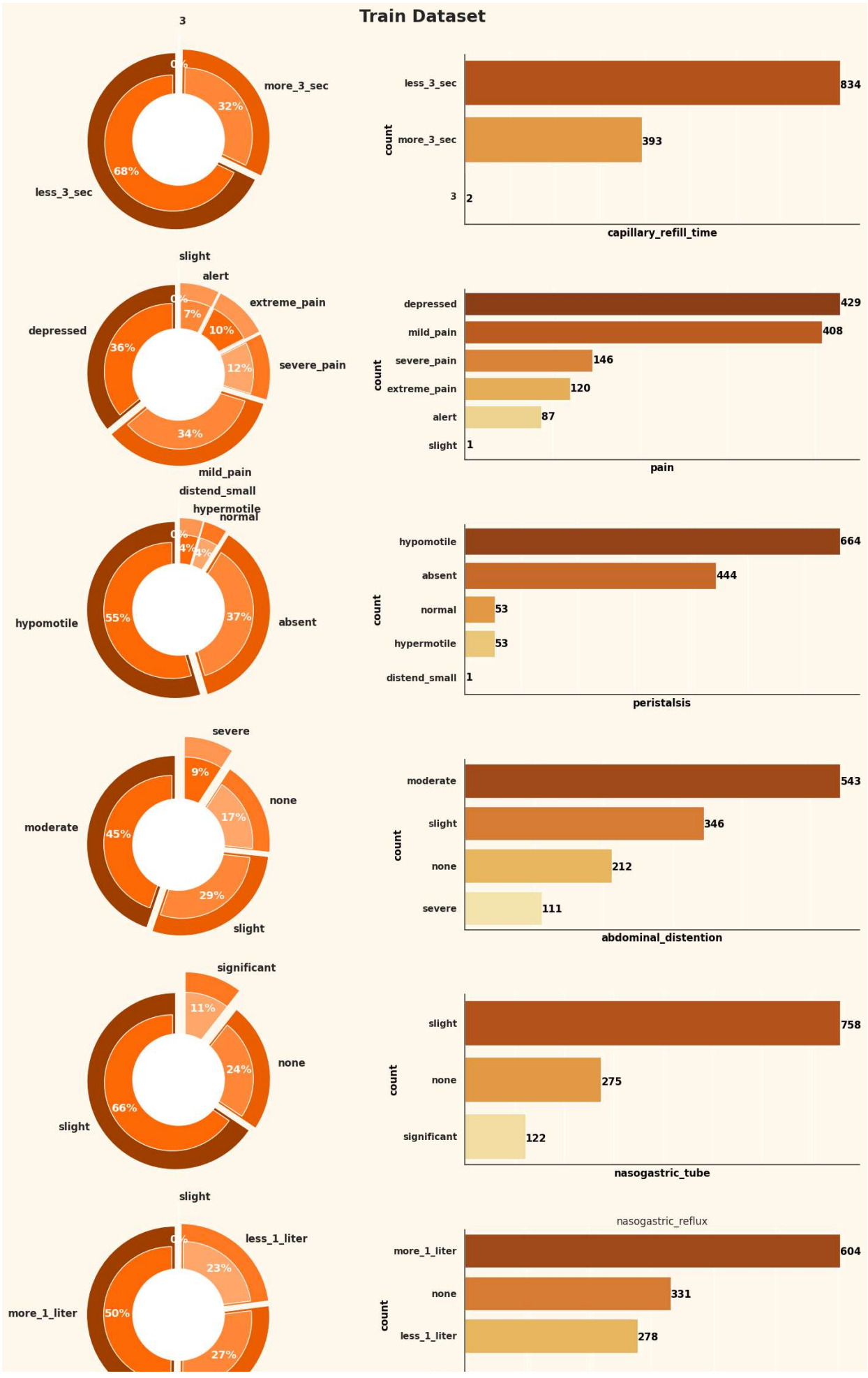
Exploratory Data Analysis

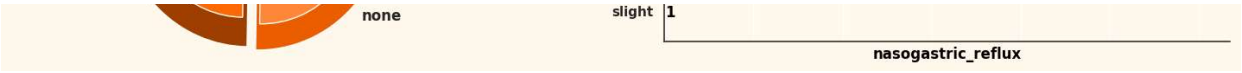
In [70]:
```
selected_columns = ['surgery',
 'age',
'temp_of_extremities',
 'peripheral_pulse',
 'mucous_membrane',
 'outcome']
plot_count(train, selected_columns)
```


Train Dataset
no
28%
72%
yes
surgery
yes
no
count
887
348
young
6%
94%
adult
age
adult
young
count
1160
75
warm
cold
4%
10%
27%
normal
59%
cool
temp_of_extremities
cool
normal
cold
warm
count
700
324
121
51
increased
absent
37%
normal
62%
reduced
peripheral_pulse
reduced
normal
absent
increased
count
724
429
18
4
bright_pink
pale_pink
7%
dark_cyanotic
23%
10%
17%
bright_red
21%
normal_pink
21%
pale_cyanotic
mucous_membrane
pale_pink
normal_pink
pale_cyanotic
bright_red
dark_cyanotic
bright_pink
count
284
260
250
211
121
88
1
20%
2
46%
outcome
2
0
count
574
410

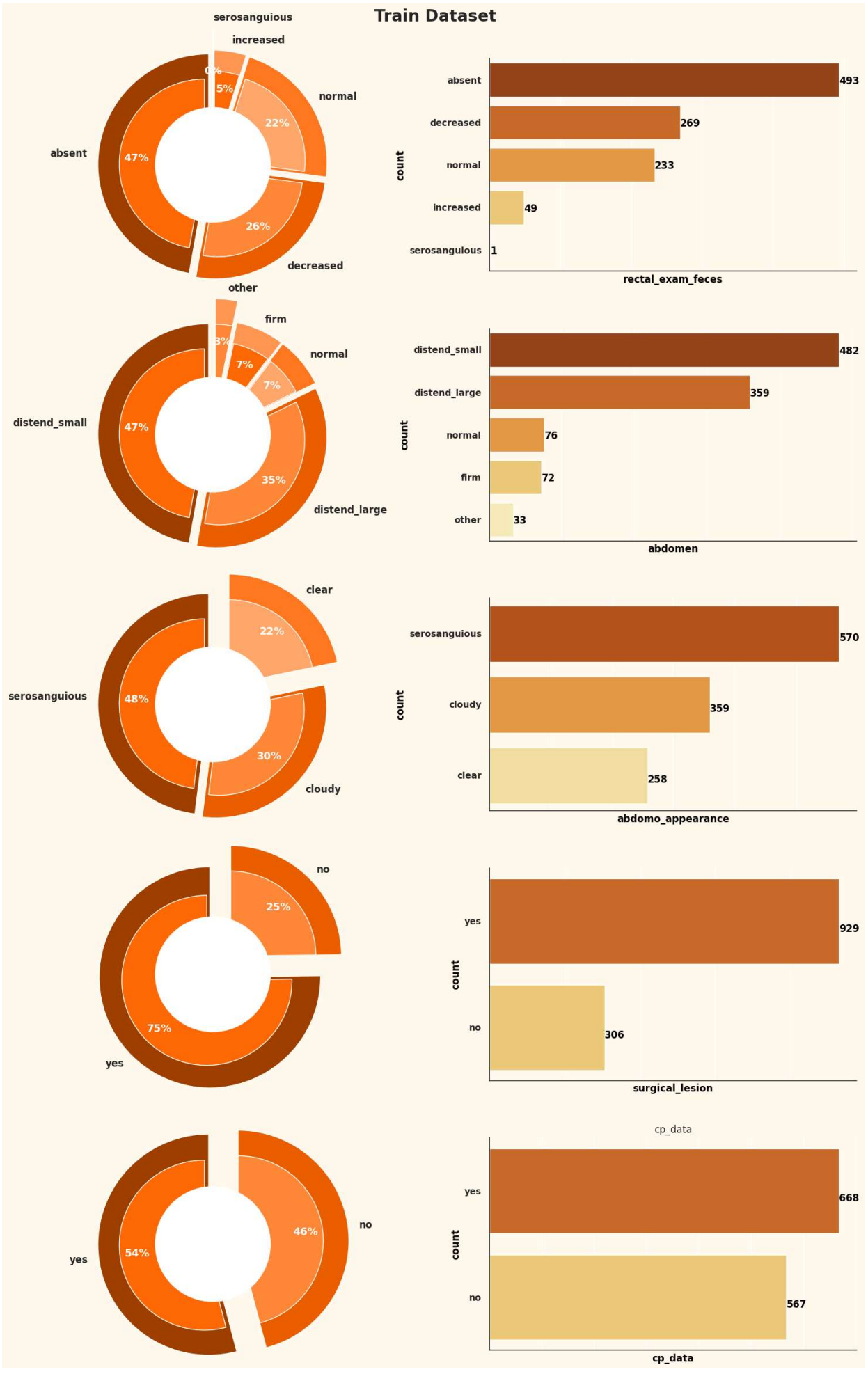


In [71]:
```
selected_columns = ['capillary_refill_time',
 'pain',
 'peristalsis',
 'abdominal_distention',
 'nasogastric_tube',
 'nasogastric_reflux']
plot_count(train, selected_columns)
```


Train Dataset
3
more_3_sec
32%
68%
less_3_sec
less_3_sec
more_3_sec
3
834
393
2
count
capillary_refill_time
slight
alert
extreme_pain
severe_pain
mild_pain
depressed
7%
10%
12%
34%
36%
depressed
mild_pain
severe_pain
extreme_pain
alert
slight
429
408
146
120
87
1
count
pain
distend_small
hypermotile
normal
absent
hypomotile
4%
4%
37%
55%
hypomotile
absent
normal
hypermotile
distend_small
664
444
53
53
1
count
peristalsis
severe
none
slight
moderate
9%
17%
29%
45%
moderate
slight
none
severe
543
346
212
111
count
abdominal_distention
significant
none
slight
11%
24%
66%
slight
none
significant
758
275
122
count
nasogastric_tube
slight
less_1_liter
more_1_liter
23%
27%
50%
nasogastric_reflux
more_1_liter
none
less_1_liter
604
331
278
count



In [72]:
```
selected_columns = ['rectal_exam_feces',
 'abdomen',
 'abdomo_appearance',
 'surgical_lesion',
 'cp_data']
plot_count(train, selected_columns)
```


Train Dataset
serosanguious
increased
normal
absent
decreased
0%
5%
22%
47%
26%
absent
decreased
normal
increased
serosanguious
493
269
233
49
1
count
rectal_exam_feces
other
firm
normal
distend_small
distend_large
3%
7%
7%
47%
35%
distend_small
distend_large
normal
firm
other
482
359
76
72
33
count
abdomen
clear
serosanguious
cloudy
22%
48%
30%
serosanguious
cloudy
clear
570
359
258
count
abdomo_appearance
no
yes
25%
75%
yes
no
929
306
count
surgical_lesion
cp_data
no
yes
46%
54%
yes
no
668
567
count
cp_data

AutoML Models Training

In [73]:
```
predictor = TabularPredictor(label='outcome', eval_metric='f1_micro',)
```

```
No path specified. Models will be saved in: "AutogluonModels/ag-20231002_051218/"
Beginning AutoGluon training ...
AutoGluon will save models to "AutogluonModels/ag-20231002_051218/"
AutoGluon Version:  0.8.2
Python Version:     3.10.12
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Sun Sep 24 13:27:25 UTC 2023
Disk Space Avail:   20.87 GB / 20.96 GB (99.6%)
Train Data Rows:    1235
Train Data Columns: 28
Label Column: outcome
Preprocessing data ...
AutoGluon infers your prediction problem is: 'multiclass' (because dtype of label-col
umn == int, but few unique label-values observed).
        3 unique label values:  [0, 1, 2]
        If 'multiclass' is not the correct problem_type, please manually specify the
problem_type parameter during predictor init (You may specify problem_type as one of:
['binary', 'multiclass', 'regression'])
Train Data Class Count: 3
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
        Available Memory:                    31508.07 MB
        Train Data (Original)  Memory Usage: 1.36 MB (0.0% of available memory)
        Inferring data type of each feature based on column values. Set feature_metad
ata_in to manually specify special dtypes of the features.
        Stage 1 Generators:
                Fitting AsTypeFeatureGenerator...
                        Note: Converting 5 features to boolean dtype as they only con
tain 2 unique values.
        Stage 2 Generators:
                Fitting FillNaFeatureGenerator...
        Stage 3 Generators:
                Fitting IdentityFeatureGenerator...
                Fitting CategoryFeatureGenerator...
                        Fitting CategoryMemoryMinimizeFeatureGenerator...
        Stage 4 Generators:
                Fitting DropUniqueFeatureGenerator...
        Stage 5 Generators:
                Fitting DropDuplicatesFeatureGenerator...
        Types of features in original data (raw dtype, special dtypes):
                ('float', [])  :  7 | ['rectal_temp', 'pulse', 'respiratory_rate', 'n
asogastric_reflux_ph', 'packed_cell_volume', ...]
                ('int', [])    :  5 | ['id', 'hospital_number', 'lesion_1', 'lesion_
2', 'lesion_3']
                ('object', []) : 16 | ['surgery', 'age', 'temp_of_extremities', 'peri
pheral_pulse', 'mucous_membrane', ...]
        Types of features in processed data (raw dtype, special dtypes):
                ('category', [])  : 12 | ['temp_of_extremities', 'peripheral_pulse',
'mucous_membrane', 'capillary_refill_time', 'pain', ...]
                ('float', [])     :  7 | ['rectal_temp', 'pulse', 'respiratory_rate',
'nasogastric_reflux_ph', 'packed_cell_volume', ...]
                ('int', [])       :  4 | ['id', 'hospital_number', 'lesion_1', 'lesio
n_2']
                ('int', ['bool']) :  5 | ['surgery', 'age', 'surgical_lesion', 'lesio
n_3', 'cp_data']
        0.2s = Fit runtime
        28 features in original data used to generate 28 features in processed data.
        Train Data (Processed) Memory Usage: 0.14 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.28s ...
```

```
AutoGluon will gauge predictive performance using evaluation metric: 'f1_micro'
        To change this, specify the eval_metric parameter of Predictor()
Automatically generating train/validation split with holdout_frac=0.2, Train Rows: 98
8, Val Rows: 247
User-specified model hyperparameters to be fit:
{
        'NN_TORCH': {},
        'GBM': [{'extra_trees': True, 'ag_args': {'name_suffix': 'XT'}}, {}, 'GBMLarg
e'],
        'CAT': {},
        'XGB': {},
        'FASTAI': {},
        'RF': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini', 'problem_type
s': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args': {'name_suffix':
'Entr', 'problem_types': ['binary', 'multiclass']}}, {'criterion': 'squared_error',
'ag_args': {'name_suffix': 'MSE', 'problem_types': ['regression', 'quantile']}}],
        'XT': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini', 'problem_type
s': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args': {'name_suffix':
'Entr', 'problem_types': ['binary', 'multiclass']}}, {'criterion': 'squared_error',
'ag_args': {'name_suffix': 'MSE', 'problem_types': ['regression', 'quantile']}}],
        'KNN': [{'weights': 'uniform', 'ag_args': {'name_suffix': 'Unif'}}, {'weight
s': 'distance', 'ag_args': {'name_suffix': 'Dist'}}],
}
Fitting 13 L1 models ...
Fitting model: KNeighborsUnif ...
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.fit: running accelerated versi
on on CPU
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.predict_proba: running acceler
ated version on CPU
        0.5547   = Validation score   (f1_micro)
        0.02s    = Training   runtime
        0.01s    = Validation runtime
Fitting model: KNeighborsDist ...
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.fit: running accelerated versi
on on CPU
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.predict_proba: running acceler
ated version on CPU
        0.5587   = Validation score   (f1_micro)
        0.01s    = Training   runtime
        0.01s    = Validation runtime
Fitting model: NeuralNetFastAI ...
        0.7045   = Validation score   (f1_micro)
        1.84s    = Training   runtime
        0.02s    = Validation runtime
Fitting model: LightGBMXT ...
        0.7287   = Validation score   (f1_micro)
        3.3s     = Training   runtime
        0.03s    = Validation runtime
Fitting model: LightGBM ...
        0.7045   = Validation score   (f1_micro)
        2.5s     = Training   runtime
        0.02s    = Validation runtime
Fitting model: RandomForestGini ...
        0.7206   = Validation score   (f1_micro)
        1.15s    = Training   runtime
        0.12s    = Validation runtime
Fitting model: RandomForestEntr ...
        0.7045   = Validation score   (f1_micro)
        1.09s    = Training   runtime
        0.13s    = Validation runtime
```

```
Fitting model: CatBoost ...
        0.7004   = Validation score   (f1_micro)
        28.39s   = Training   runtime
        0.01s    = Validation runtime
Fitting model: ExtraTreesGini ...
        0.664    = Validation score   (f1_micro)
        0.94s    = Training   runtime
        0.12s    = Validation runtime
Fitting model: ExtraTreesEntr ...
        0.668    = Validation score   (f1_micro)
        1.06s    = Training   runtime
        0.12s    = Validation runtime
Fitting model: XGBoost ...
        0.7409   = Validation score   (f1_micro)
        2.0s     = Training   runtime
        0.02s    = Validation runtime
Fitting model: NeuralNetTorch ...
        0.6802   = Validation score   (f1_micro)
        4.62s    = Training   runtime
        0.02s    = Validation runtime
Fitting model: LightGBMLarge ...
        0.7085   = Validation score   (f1_micro)
        6.26s    = Training   runtime
        0.04s    = Validation runtime
Fitting model: WeightedEnsemble_L2 ...
        0.7409   = Validation score   (f1_micro)
        2.03s    = Training   runtime
        0.0s     = Validation runtime
AutoGluon training complete, total runtime = 57.24s ... Best model: "WeightedEnsemble
_L2"
TabularPredictor saved. To load, use: predictor = TabularPredictor.load("AutogluonMod
els/ag-20231002_051218/")
```

Show the result comparison on the different ML models

In [74]:
```
ld_board
```

```
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.predict_proba: running acceler
ated version on CPU
INFO:sklearnex: sklearn.neighbors.KNeighborsClassifier.predict_proba: running acceler
ated version on CPU
```

Out[74]:

	model	score_test	score_val	pred_time_test	pred_time_val	fit_time	pred_time_te
0	XGBoost	0.948178	0.740891	0.072770	0.017220	1.995450	
1	WeightedEnsemble_L2	0.948178	0.740891	0.077072	0.018997	4.022708	
2	LightGBMXT	0.944130	0.728745	0.122196	0.025876	3.298457	
3	RandomForestGini	0.944130	0.720648	0.181190	0.122222	1.148319	
4	LightGBMLarge	0.941700	0.708502	0.252974	0.041560	6.264603	
5	LightGBM	0.940891	0.704453	0.040893	0.016327	2.496833	
6	RandomForestEntr	0.940891	0.704453	0.178833	0.134928	1.089041	
7	ExtraTreesEntr	0.933603	0.668016	0.181032	0.118712	1.057424	
8	ExtraTreesGini	0.932794	0.663968	0.185565	0.120928	0.944986	
9	NeuralNetFastAI	0.916599	0.704453	0.050212	0.020880	1.839486	
10	KNeighborsDist	0.905263	0.558704	0.010698	0.007454	0.012149	
11	CatBoost	0.877733	0.700405	0.021446	0.012062	28.394323	
12	NeuralNetTorch	0.791903	0.680162	0.034004	0.021040	4.616868	
13	KNeighborsUnif	0.675304	0.554656	0.010478	0.008425	0.016792	

Predict

In [75]:
```
pred_y = predictor.predict_proba(test)
pred_y
```

Out[75]:

	0	1	2
0	0.006877	0.012475	0.980648
1	0.349543	0.050058	0.600400
2	0.011721	0.005000	0.983279
3	0.023709	0.950045	0.026246
4	0.032240	0.025802	0.941958
...	...	...	...
819	0.786331	0.111169	0.102500
820	0.852056	0.109158	0.038786
821	0.554046	0.038098	0.407856
822	0.007001	0.003387	0.989612
823	0.094274	0.021617	0.884108

824 rows × 3 columns

In [76]:
```
pred = predictor.predict(test)
pred
```

Out[76]:
```
0      2
1      2
2      2
3      1
4      2
      ..
819    0
820    0
821    0
822    2
823    2
Name: outcome, Length: 824, dtype: int64
```

Generating the Submission File

In [77]:
```
sub.head()
```

Out[77]:

	id	outcome
0	1235	lived
1	1236	lived
2	1237	lived
3	1238	lived
4	1239	lived

In [78]:
```
submit = pd.DataFrame()

submit['id'] = test['id']
submit['outcome'] = pred
```

In [79]:
```
submit["outcome"] = submit["outcome"].map({0:"died", 1:"euthanized", 2:"lived"})
```

In [80]:
```
submit.head()
```

Out[80]:

	id	outcome
0	1235	lived
1	1236	lived
2	1237	lived
3	1238	euthanized
4	1239	lived

In [81]:
```
submit.to_csv('My Submission .csv', index=False)
```