In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
import pandas as pd
```

In [3]:

```
import numpy as np
```

In [4]:

```
from sklearn.model_selection import train_test_split,cross_val_score
```

In [5]:

```
from sklearn.metrics import accuracy_score
```

In [6]:

```
from sklearn.utils import shuffle
```

In [7]:

```
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
```

In [8]:

```
from sklearn.pipeline import Pipeline
```

In [9]:

```
from sklearn.metrics import classification_report
```

In [10]:

```
from sklearn.covariance import EllipticEnvelope
```

In [11]:

```
import matplotlib.pyplot as plt
```

In [12]:

```
import seaborn as sns
```

In [13]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

```
h30hz0  = pd.read_csv("h30hz0.csv")
```

In [16]:

```
h30hz10 = pd.read_csv("h30hz10.csv")
```

In [17]:

```
h30hz20 = pd.read_csv("h30hz20.csv")
```

In [18]:

```
h30hz30 = pd.read_csv("h30hz30.csv")
```

In [19]:

```
h30hz40 = pd.read_csv("h30hz40.csv")
```

In [20]:

```
h30hz50 = pd.read_csv("h30hz50.csv")
```

In [21]:

```
h30hz60 = pd.read_csv("h30hz60.csv")
```

In [22]:

```
h30hz70 = pd.read_csv("h30hz70.csv")
```

In [23]:

```
h30hz80 = pd.read_csv("h30hz80.csv")
```

In [24]:

```
h30hz90 = pd.read_csv("h30hz90.csv")
```

In [25]:

```
b30hz0  = pd.read_csv("b30hz0.csv")
```

In [26]:

```
b30hz10 = pd.read_csv("b30hz10.csv")
```

In [27]:

```
b30hz20 = pd.read_csv("b30hz20.csv")
```

In [28]:

```
b30hz30 = pd.read_csv("b30hz30.csv")
```

In [29]:

```
b30hz40 = pd.read_csv("b30hz40.csv")
```

In [30]:

```
b30hz50 = pd.read_csv("b30hz50.csv")
```

In [31]:

```
b30hz60 = pd.read_csv("b30hz60.csv")
```

In [32]:

```
b30hz70 = pd.read_csv("b30hz70.csv")
```

In [33]:

```
b30hz80 = pd.read_csv("b30hz80.csv")
```

In [34]:

```
b30hz90 = pd.read_csv("b30hz90.csv")
```

In [35]:

```
failure = 0
load = 0
```

In [36]:

```
h30hz0['load'] = load*np.ones((len(h30hz0.index),1))
failureArray = np.zeros((len(h30hz0.index),1))
h30hz0['failure'] = failureArray
```

In [37]:

```
load = 10

h30hz10['load'] = load*np.ones((len(h30hz10.index),1))
failureArray = np.zeros((len(h30hz10.index),1))
h30hz10['failure'] = failureArray
```

In [38]:

```
load = 20

h30hz20['load'] = load*np.ones((len(h30hz20.index),1))
failureArray = np.zeros((len(h30hz20.index),1))
h30hz20['failure'] = failureArray
```

In [39]:

```
load = 30

h30hz30['load'] = load*np.ones((len(h30hz30.index),1))
failureArray = np.zeros((len(h30hz30.index),1))
h30hz30['failure'] = failureArray
```

In [40]:

```
load = 40

h30hz40['load'] = load*np.ones((len(h30hz40.index),1))
failureArray = np.zeros((len(h30hz40.index),1))
h30hz40['failure'] = failureArray
```

In [41]:

```
load = 50

h30hz50['load'] = load*np.ones((len(h30hz50.index),1))
failureArray = np.zeros((len(h30hz50.index),1))
h30hz50['failure'] = failureArray
```

In [42]:

```
load = 60

h30hz60['load'] = load*np.ones((len(h30hz60.index),1))
failureArray = np.zeros((len(h30hz60.index),1))
h30hz60['failure'] = failureArray
```

In [43]:

```

load = 70

h30hz70['load'] = load*np.ones((len(h30hz70.index),1))
failureArray = np.zeros((len(h30hz70.index),1))
h30hz70['failure'] = failureArray
```

In [44]:

```
load = 80

h30hz80['load'] = load*np.ones((len(h30hz80.index),1))
failureArray = np.zeros((len(h30hz80.index),1))
h30hz80['failure'] = failureArray
```

In [45]:

```
load = 90

h30hz90['load'] = load*np.ones((len(h30hz90.index),1))
failureArray = np.zeros((len(h30hz90.index),1))
h30hz90['failure'] = failureArray
```

In [46]:

```
failure = 1

load = 0
```

In [47]:

```
b30hz0['load'] = load*np.ones((len(b30hz0.index),1))
failureArray = np.ones((len(b30hz0.index),1))
b30hz0['failure'] = failureArray
```

In [48]:

```
load = 10

b30hz10['load'] = load*np.ones((len(b30hz10.index),1))
failureArray = np.ones((len(b30hz10.index),1))
b30hz10['failure'] = failureArray
```

In [49]:

```
load = 20

b30hz20['load'] = load*np.ones((len(b30hz20.index),1))
failureArray = np.ones((len(b30hz20.index),1))
b30hz20['failure'] = failureArray
```

In [50]:

```
load = 30

b30hz30['load'] = load*np.ones((len(b30hz30.index),1))
failureArray = np.ones((len(b30hz30.index),1))
b30hz30['failure'] = failureArray
```

In [51]:

```
load = 40

b30hz40['load'] = load*np.ones((len(b30hz40.index),1))
failureArray = np.ones((len(b30hz40.index),1))
b30hz40['failure'] = failureArray
```

In [52]:

```
load = 50

b30hz50['load'] = load*np.ones((len(b30hz50.index),1))
failureArray = np.ones((len(b30hz50.index),1))
b30hz50['failure'] = failureArray
```

In [53]:

```
load = 60

b30hz60['load'] = load*np.ones((len(b30hz60.index),1))
failureArray = np.ones((len(b30hz60.index),1))
b30hz60['failure'] = failureArray
```

In [54]:

```
load = 70

b30hz70['load'] = load*np.ones((len(b30hz70.index),1))
failureArray = np.ones((len(b30hz70.index),1))
b30hz70['failure'] = failureArray
```

In [55]:

```
load = 80

b30hz80['load'] = load*np.ones((len(b30hz80.index),1))
failureArray = np.ones((len(b30hz80.index),1))
b30hz80['failure'] = failureArray
```

In [56]:

```
load = 90

b30hz90['load'] = load*np.ones((len(b30hz90.index),1))
failureArray = np.ones((len(b30hz90.index),1))
b30hz90['failure'] = failureArray
```

In [57]:

```
broken_df = pd.concat([b30hz0,b30hz10,b30hz20,b30hz30,b30hz40,b30hz50,b30hz60,b
```

In [58]:

```
healthy_df = pd.concat([h30hz0,h30hz10,h30hz20,h30hz30,h30hz40,h30hz50,h30hz60,
```

In [59]:

```
gear_data   = pd.concat([broken_df,healthy_df], axis =0)
```

In [60]:

```
gear_data.head()
```

Out[60]:

	a1	a2	a3	a4	load	failure
0	2.350390	1.454870	-1.667080	-2.055610	0.0	1.0
1	2.452970	1.400100	-2.825100	0.984487	0.0	1.0
2	-0.241284	-0.267390	0.793540	0.605862	0.0	1.0
3	1.130270	-0.890918	0.696969	0.613068	0.0	1.0
4	-1.296140	0.980479	-1.130560	-0.346971	0.0	1.0

In [61]:

```
training_features = ['a1', 'a2', 'a3', 'a4']
label = ['failure']
```

In [62]:

```
x = gear_data[training_features]
x.shape
```

Out[62]:

```
(2021119, 4)
```

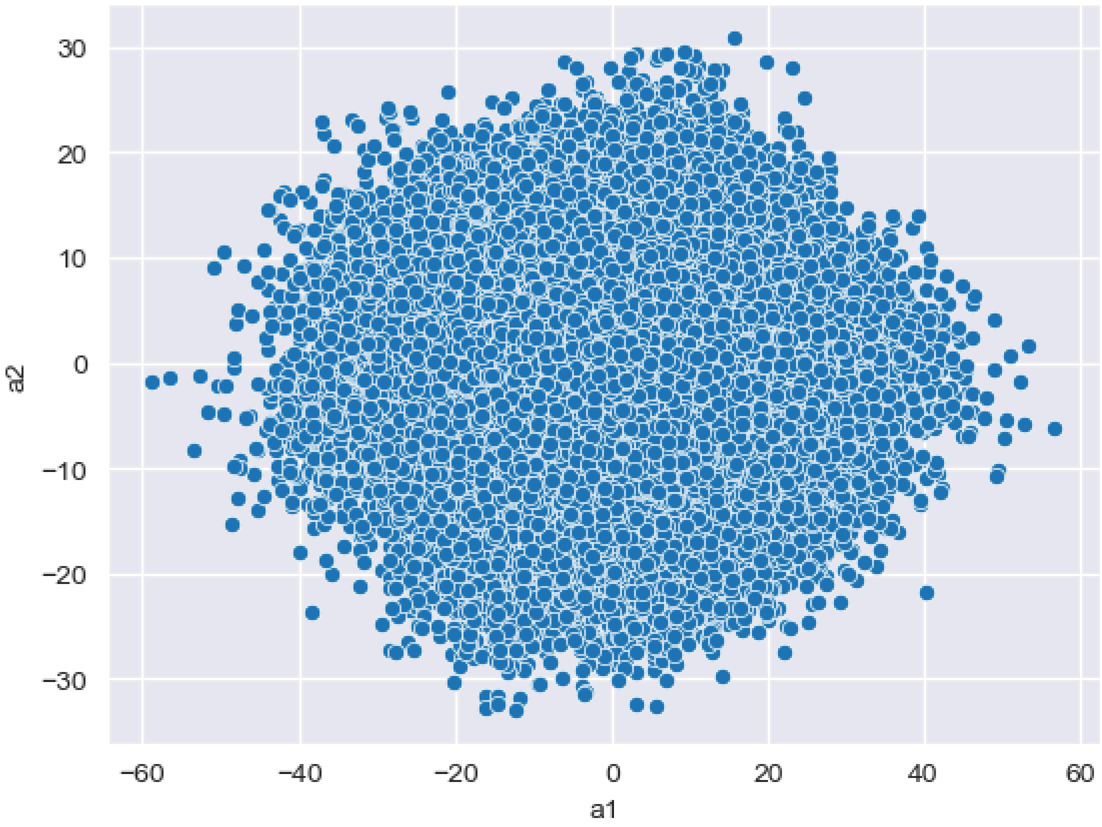
```
In [63]: y = gear_data[label]
         y.shape

Out[63]: (2021119, 1)
```

```
In [64]: X,y = shuffle(x,y)
```

```
In [65]: X.head()

Out[65]:
```

	a1	a2	a3	a4
23585	16.012000	-4.702820	-2.12148	2.25368
256255	-12.613000	-0.079185	-4.19099	4.61565
596712	-0.499146	1.158620	1.59286	1.55654
695881	-10.388100	7.121680	-3.70214	4.60284
771571	-2.198550	2.512490	5.43726	-3.96574

```
In [66]: sns.set_style("darkgrid")
         sns.scatterplot(x=X['a1'][:], y=X['a2'][:])

Out[66]: <Axes: xlabel='a1', ylabel='a2'>
```

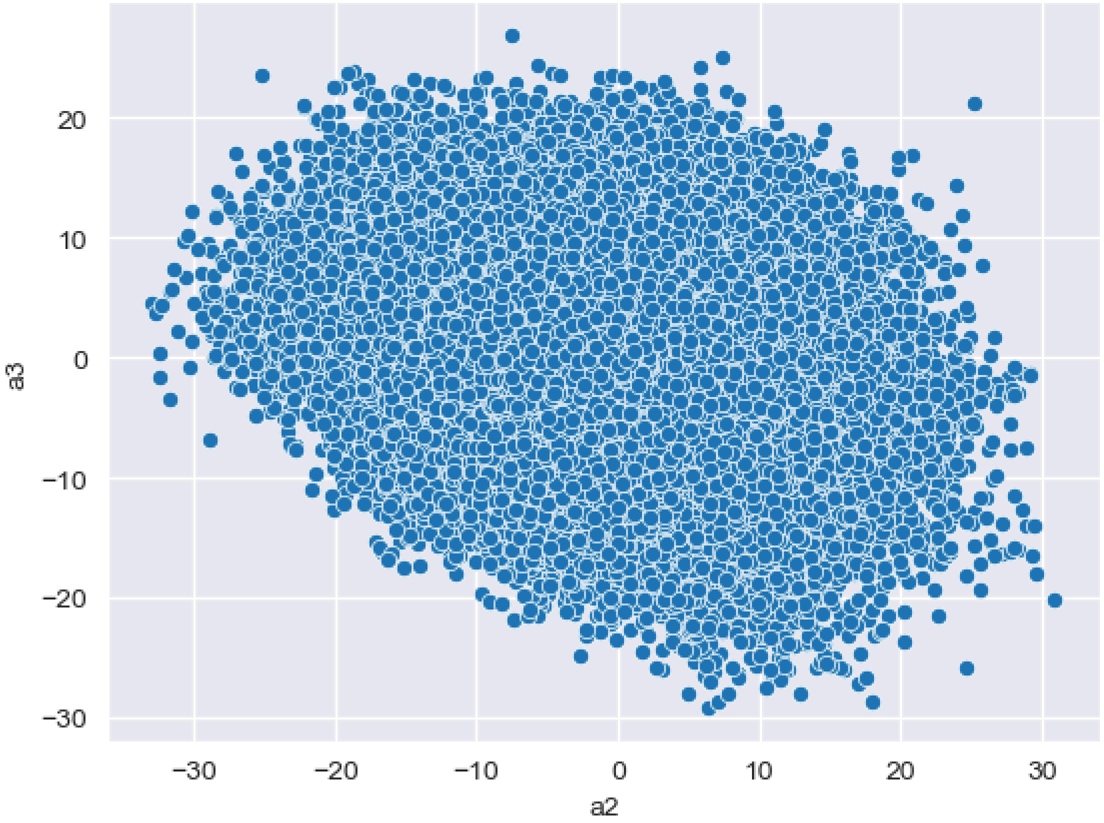


In [67]:

```
sns.set_style("darkgrid")
sns.scatterplot(x=X['a2'][:], y=X['a3'][:])
```

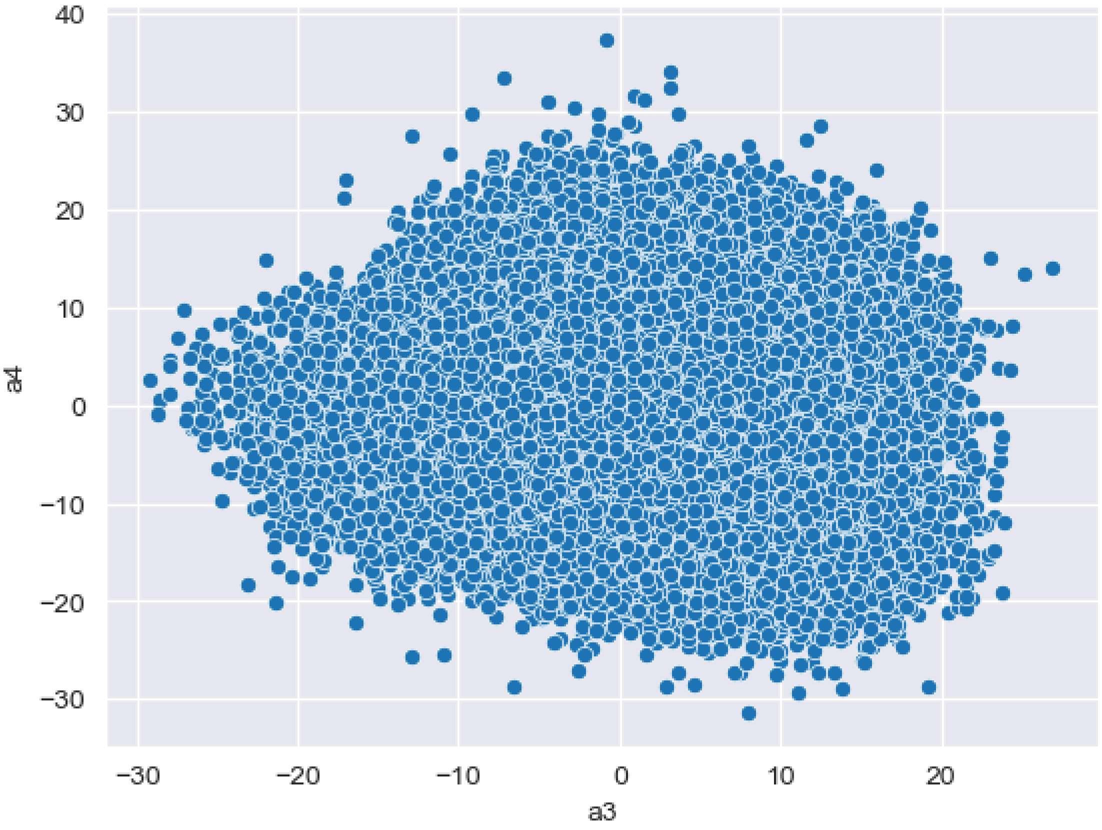
Out[67]:

```
<Axes: xlabel='a2', ylabel='a3'>
```



```
In [68]: sns.set_style("darkgrid")
         sns.scatterplot(x=X['a3'][:], y=X['a4'][:])
```

```
Out[68]: <Axes: xlabel='a3', ylabel='a4'>
```



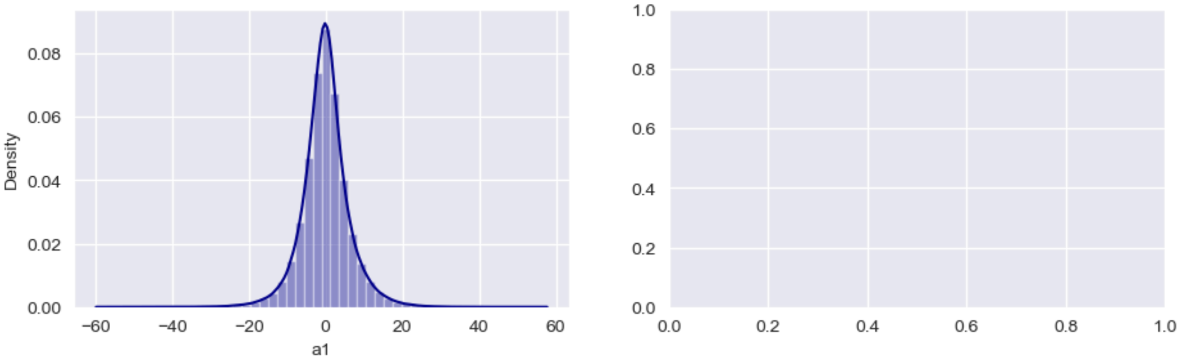
In [69]:

```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(11,3))
sns.distplot(X['a1'][:], ax=ax[0], color="darkblue")
```

```
C:\Users\Manic\AppData\Local\Temp\ipykernel_11212\2442753647.py:2: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(X['a1'][:], ax=ax[0], color="darkblue")
```

Out[69]:

```
<Axes: xlabel='a1', ylabel='Density'>
```



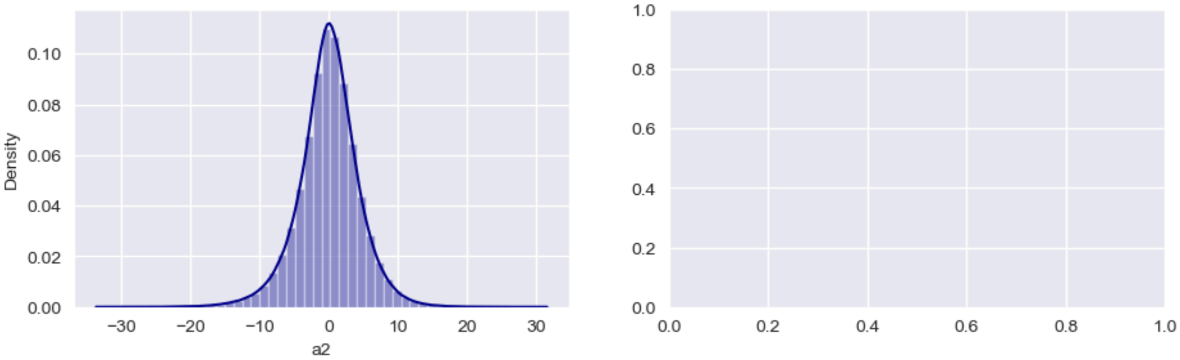
In [70]:

```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(11,3))
sns.distplot(X['a2'][:], ax=ax[0], color="darkblue")
```

```
C:\Users\Manic\AppData\Local\Temp\ipykernel_11212\816255312.py:2: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(X['a2'][:], ax=ax[0], color="darkblue")
```

Out[70]:

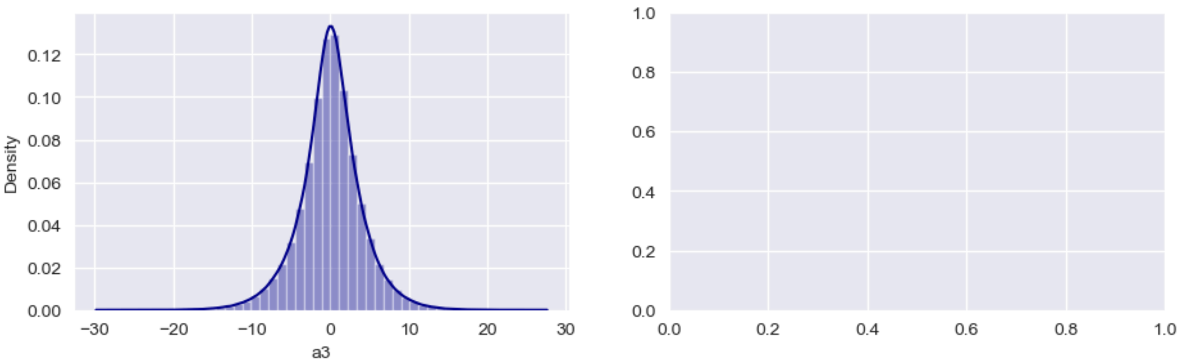
```
<Axes: xlabel='a2', ylabel='Density'>
```



In [71]:
```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(11,3))
sns.distplot(X['a3'][:], ax=ax[0], color="darkblue")
```

```
C:\Users\Manic\AppData\Local\Temp\ipykernel_11212\3235081541.py:2: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(X['a3'][:], ax=ax[0], color="darkblue")
```

Out[71]:
```
<Axes: xlabel='a3', ylabel='Density'>
```



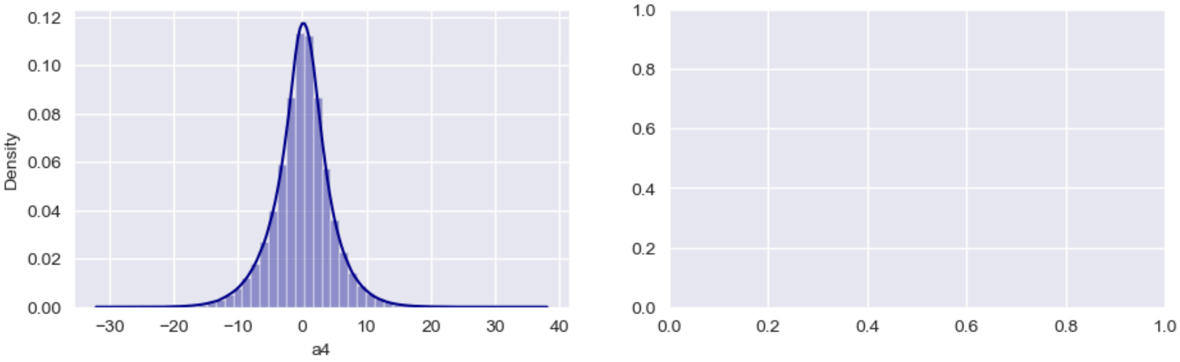
In [72]:

```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(11,3))
sns.distplot(X['a4'][:], ax=ax[0], color="darkblue")
```

```
C:\Users\Manic\AppData\Local\Temp\ipykernel_11212\1657850364.py:2: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(X['a4'][:], ax=ax[0], color="darkblue")
```

Out[72]:

```
<Axes: xlabel='a4', ylabel='Density'>
```



In [73]:

```
elpenv = EllipticEnvelope(contamination=0.1, random_state=2)
```

In [74]:

```
pred = elpenv.fit_predict(X.iloc[:500])
```

In [75]:

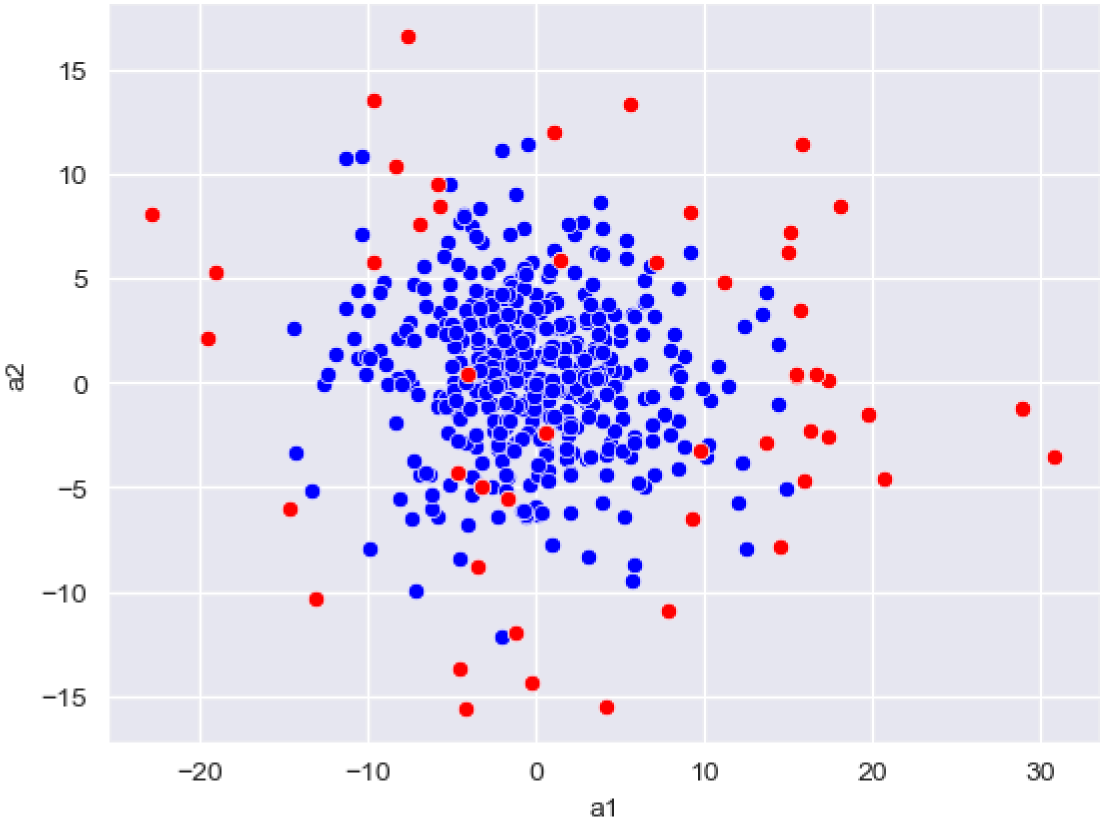
```
outlier_index = np.where(pred==-1)
outlier_index
outlier_values = X.iloc[outlier_index]
```

In [76]:

```
sns.scatterplot(x=X['a1'][:500], y=X['a2'][:500], color = 'b')
sns.scatterplot(x=outlier_values['a1'][:500],y=outlier_values['a2'][:500], colo
```

Out[76]:

```
<Axes: xlabel='a1', ylabel='a2'>
```



In [77]:

```
# Red colors are outliers
```

In [ ]: