In [2]:

```
pip install mpld3
```

```
Collecting mpld3
Installing collected packages: mpld3
Successfully installed mpld3-0.3
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.rcParams["figure.figsize"] = (10, 20)
import mpld3
mpld3.enable_notebook()
```

In [15]:

```
pd.set_option('display.max_columns', 500)
df = pd.read_csv('oil_spill_csv.csv')
```

In [16]:

```
df.head()
```

Out[16]:

| | class | attr1 | attr2 | attr3 | attr4 | attr5 | attr6 | attr7 | attr8 | attr9 | attr10 | attr11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | '1' | 1 | 2558 | 1506.09 | 456.63 | 90 | 6395000.0 | 40.88 | 7.89 | 29780.0 | 0.19 | 214.7 |
| 1 | '-1' | 2 | 22325 | 79.11 | 841.03 | 180 | 55812500.0 | 51.11 | 1.21 | 61900.0 | 0.02 | 901.7 |
| 2 | '1' | 3 | 115 | 1449.85 | 608.43 | 88 | 287500.0 | 40.42 | 7.34 | 3340.0 | 0.18 | 86.1 |
| 3 | '1' | 4 | 1201 | 1562.53 | 295.65 | 66 | 3002500.0 | 42.40 | 7.97 | 18030.0 | 0.19 | 166.5 |
| 4 | '-1' | 5 | 312 | 950.27 | 440.86 | 37 | 780000.0 | 41.43 | 7.03 | 3350.0 | 0.17 | 232.8 |

In [17]:

```
df.isnull().sum()
```

Out[17]:

```
class     0
attr1     0
attr2     0
attr3     0
attr4     0
attr5     0
attr6     0
attr7     0
attr8     0
attr9     0
attr10    0
attr11    0
attr12    0
attr13    0
attr14    0
attr15    0
attr16    0
attr17    0
attr18    0
attr19    0
attr20    0
attr21    0
attr22    0
attr23    0
attr24    0
attr25    0
attr26    0
attr27    0
attr28    0
attr29    0
attr30    0
attr31    0
attr32    0
attr33    0
attr34    0
attr35    0
attr36    0
attr37    0
attr38    0
attr39    0
attr40    0
attr41    0
attr42    0
attr43    0
attr44    0
attr45    0
attr46    0
attr47    0
attr48    0
attr49    0
dtype: int64
```

In [18]:

```
df.shape
```

Out[18]:

```
(937, 50)
```

In [19]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 937 entries, 0 to 936
Data columns (total 50 columns):
class     937 non-null object
attr1     937 non-null int64
attr2     937 non-null int64
attr3     937 non-null float64
attr4     937 non-null float64
attr5     937 non-null int64
attr6     937 non-null float64
attr7     937 non-null float64
attr8     937 non-null float64
attr9     937 non-null float64
attr10    937 non-null float64
attr11    937 non-null float64
attr12    937 non-null float64
attr13    937 non-null float64
attr14    937 non-null float64
attr15    937 non-null float64
attr16    937 non-null float64
attr17    937 non-null float64
attr18    937 non-null float64
attr19    937 non-null float64
attr20    937 non-null float64
attr21    937 non-null float64
attr22    937 non-null float64
attr23    937 non-null int64
attr24    937 non-null float64
attr25    937 non-null float64
attr26    937 non-null float64
attr27    937 non-null float64
attr28    937 non-null float64
attr29    937 non-null float64
attr30    937 non-null float64
attr31    937 non-null float64
attr32    937 non-null float64
attr33    937 non-null float64
attr34    937 non-null float64
attr35    937 non-null int64
attr36    937 non-null int64
attr37    937 non-null float64
attr38    937 non-null float64
attr39    937 non-null int64
attr40    937 non-null int64
attr41    937 non-null float64
attr42    937 non-null float64
attr43    937 non-null float64
attr44    937 non-null float64
attr45    937 non-null float64
attr46    937 non-null int64
attr47    937 non-null float64
attr48    937 non-null float64
attr49    937 non-null float64
dtypes: float64(40), int64(9), object(1)
memory usage: 366.1+ KB
```

In [20]:

```
df.dtypes
```

Out[20]:

```
class      object
attr1       int64
attr2       int64
attr3     float64
attr4     float64
attr5       int64
attr6     float64
attr7     float64
attr8     float64
attr9     float64
attr10    float64
attr11    float64
attr12    float64
attr13    float64
attr14    float64
attr15    float64
attr16    float64
attr17    float64
attr18    float64
attr19    float64
attr20    float64
attr21    float64
attr22    float64
attr23      int64
attr24    float64
attr25    float64
attr26    float64
attr27    float64
attr28    float64
attr29    float64
attr30    float64
attr31    float64
attr32    float64
attr33    float64
attr34    float64
attr35      int64
attr36      int64
attr37    float64
attr38    float64
attr39      int64
attr40      int64
attr41    float64
attr42    float64
attr43    float64
attr44    float64
attr45    float64
attr46      int64
attr47    float64
attr48    float64
attr49    float64
dtype: object
```

In [21]:

```
sns.heatmap(df.isnull(), yticklabels = False, cbar = False, cmap = 'viridis')
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27692eb8a88>
```

In [22]:

```
sns.pairplot(data = df, hue = 'class')
```

```
C:\Users\SURYA\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.p
y:487: RuntimeWarning: invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Users\SURYA\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdeto
ols.py:34: RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```

Out[22]:

```
<seaborn.axisgrid.PairGrid at 0x27692ea7648>
```

In [23]:

```
df['class'].value_counts()
```

Out[23]:

```
'-1'    896
'1'      41
Name: class, dtype: int64
```

In [24]:

```
X = df.drop(['class'], axis = 1)
y = df['class']
```

In [25]:

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

In [26]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [27]:

```
from sklearn.linear_model import LogisticRegression
```

In [28]:

```
model = LogisticRegression()
```

In [30]:

```
model.fit(X_train, y_train)
```

```
C:\Users\SURYA\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
C:\Users\SURYA\Anaconda3\lib\site-packages\sklearn\svm\base.py:929: Converge
nceWarning: Liblinear failed to converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

Out[30]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [31]:

```
predict = model.predict(X_test)
```

In [32]:

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

In [33]:

```
print(classification_report(predict, y_test))
```

```
              precision    recall  f1-score   support

        '-1'       0.99      0.96      0.98       183
         '1'       0.30      0.60      0.40         5

    accuracy                           0.95       188
   macro avg       0.64      0.78      0.69       188
weighted avg       0.97      0.95      0.96       188
```

In [34]:

```
print(accuracy_score(predict, y_test)*100)
```

```
95.2127659574468
```

In [ ]:

```

```