Prepare Data For Machine Learning (Pinnacle)

Curate and Prepare Data for various Pinnacle Use cases

In [1]:
```
# from IPython.display import Image, HTML
import os
import numpy as np
import math
import pandas as pd
import datetime
from glob import glob

import warnings
warnings.filterwarnings("ignore")           # Suppress Warning
```

Prepare Data for Use Case: Classify Suspicious Activity from AIS Data

In [2]:
```
# s3://vault-data-corpus/vessel data/H2O Generated Data/
WorkingFolder = "/Users/cv0361/Desktop/TechChallenge/Data/csv/H2O Generated Data/"

OutputDir = WorkingFolder
```

In [3]:
```
Train = pd.read_csv(WorkingFolder + "train_with_features.csv", sep=",")
Train.head()
```

Out[3]:

	X_1	Y_1	SOG_1	COG_1	Heading_1	ROT_1	Status_1	ReceiverType_1	ReceiverID_1	dataset_1	...	delta_COG_34	delta_COG_14
0	-120.000202	34.238633	20.5	285.600010	286	0	0	r	11SBARB1	2014	...	0.299990	-42.899990
1	-122.912663	47.097593	1.9	189.500000	511	128	9	b	003669987	2012	...	-9.399990	14.000000
2	-123.215730	48.770813	0.0	288.000000	274	0	0	b	003669704	2010	...	0.000000	-29.000000
3	-121.785107	36.803988	0.0	0.000000	511	0	0	r	11SMON1	2012	...	26.200000	-124.000000
4	-123.292027	48.876247	5.1	59.299999	511	0	0	b	003669705	2013	...	2.200001	-2.299999

5 rows × 75 columns

In [4]:
```
Test = pd.read_csv(WorkingFolder + "valid_short_with_features.csv", sep=",")
Test.head()
```

Out[4]:

	X_1	Y_1	SOG_1	COG_1	Heading_1	ROT_1	Status_1	ReceiverType_1	ReceiverID_1	dataset_1	...	delta_COG_34	delta_COG_14
0	-120.000202	34.238633	20.5	285.600010	286	0	0	r	11SBARB1	2014	...	0.299990	-42.899990
1	-122.912663	47.097593	1.9	189.500000	511	128	9	b	003669987	2012	...	-9.399990	14.000000
2	-123.215730	48.770813	0.0	288.000000	274	0	0	b	003669704	2010	...	0.000000	-29.000000
3	-121.785107	36.803988	0.0	0.000000	511	0	0	r	11SMON1	2012	...	26.200000	-124.000000
4	-123.292027	48.876247	5.1	59.299999	511	0	0	b	003669705	2013	...	2.200001	-2.299999

5 rows × 75 columns

In [5]:
```
# Test.columns
```

In [6]:
```
Header = ['target', 'X_1', 'Y_1', 'SOG_1', 'COG_1', 'Heading_1', 'ROT_1', 'Status_1',
       'ReceiverType_1', 'ReceiverID_1', 'dataset_1', 'time_gap_1', 'X_2',
       'Y_2', 'SOG_2', 'COG_2', 'Heading_2', 'ROT_2', 'Status_2',
       'ReceiverType_2', 'ReceiverID_2', 'time_gap_2', 'X_3', 'Y_3', 'SOG_3',
       'COG_3', 'Heading_3', 'ROT_3', 'Status_3', 'ReceiverType_3',
       'ReceiverID_3', 'time_gap_3', 'X_4', 'Y_4', 'SOG_4', 'COG_4',
       'Heading_4', 'ROT_4', 'Status_4', 'ReceiverType_4', 'ReceiverID_4',
       'time_gap_4', 'VesselType', 'Length',
       'Width', 'distance_12', 'distance_23', 'distance_34',
       'distance_14', 'speed_12', 'speed_23', 'speed_34', 'delta_speed_12',
       'delta_speed_23', 'delta_speed_34', 'delta_SOG_12', 'delta_SOG_23',
       'delta_SOG_34', 'delta_SOG_14', 'delta_COG_12', 'delta_COG_23',
       'delta_COG_34', 'delta_COG_14', 'delta_COG_heading_1',
       'delta_COG_heading_2', 'delta_COG_heading_3', 'delta_COG_heading_4',
       'dim_1', 'dim_2', 'dim_3', 'dim_4']
```

In [7]:
```
df = pd.concat([Train, Test], ignore_index=True)
df = df[Header].reset_index()
```

In [8]:
```
df
```

Out[8]:

	index	target	X_1	Y_1	SOG_1	COG_1	Heading_1	ROT_1	Status_1	ReceiverType_1	...	delta_COG_34	delta_COG_14	delta_
0	0	0	-120.000202	34.238633	20.5	285.600010	286	0	0	r	...	0.299990	-42.899990	
1	1	1	-122.912663	47.097593	1.9	189.500000	511	128	9	b	...	-9.399990	14.000000	
2	2	0	-123.215730	48.770813	0.0	288.000000	274	0	0	b	...	0.000000	-29.000000	
3	3	0	-121.785107	36.803988	0.0	0.000000	511	0	0	r	...	26.200000	-124.000000	
4	4	0	-123.292027	48.876247	5.1	59.299999	511	0	0	b	...	2.200001	-2.299999	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
11142	11142	0	-121.335513	37.950238	0.0	354.500000	511	128	15	b	...	-1.100010	39.500000	
11143	11143	0	-121.379127	37.977662	0.0	127.300000	511	0	0	b	...	-67.500004	28.999997	
11144	11144	0	-122.382120	37.763187	0.0	116.400000	184	0	0	b	...	-0.200000	8.700000	
11145	11145	0	-122.367745	37.816272	5.1	57.799999	511	0	0	b	...	0.299980	-239.800011	
11146	11146	0	-122.299983	37.793380	0.0	45.099998	273	0	0	r	...	0.200010	-258.900002	

11147 rows × 72 columns

In [9]:
```
df = df.replace([np.inf, -np.inf], np.nan)
df = df.fillna(-9999)
df
```

Out[9]:

	index	target	X_1	Y_1	SOG_1	COG_1	Heading_1	ROT_1	Status_1	ReceiverType_1	...	delta_COG_34	delta_COG_14	delta_
0	0	0	-120.000202	34.238633	20.5	285.600010	286	0	0	r	...	0.299990	-42.899990	
1	1	1	-122.912663	47.097593	1.9	189.500000	511	128	9	b	...	-9.399990	14.000000	
2	2	0	-123.215730	48.770813	0.0	288.000000	274	0	0	b	...	0.000000	-29.000000	
3	3	0	-121.785107	36.803988	0.0	0.000000	511	0	0	r	...	26.200000	-124.000000	
4	4	0	-123.292027	48.876247	5.1	59.299999	511	0	0	b	...	2.200001	-2.299999	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
11142	11142	0	-121.335513	37.950238	0.0	354.500000	511	128	15	b	...	-1.100010	39.500000	
11143	11143	0	-121.379127	37.977662	0.0	127.300000	511	0	0	b	...	-67.500004	28.999997	
11144	11144	0	-122.382120	37.763187	0.0	116.400000	184	0	0	b	...	-0.200000	8.700000	
11145	11145	0	-122.367745	37.816272	5.1	57.799999	511	0	0	b	...	0.299980	-239.800011	
11146	11146	0	-122.299983	37.793380	0.0	45.099998	273	0	0	r	...	0.200010	-258.900002	

11147 rows × 72 columns

In [10]:
```
# Output Stat Data
df.to_csv(OutputDir + "AIS_Classify_Suspicious.csv", index=None, header = True)
```

Prepare Data for Use Case: Classify Suspicious Activity from AIS Data

In [11]:
```
# s3://vault-data-corpus/vessel data/ConsolidatedAIS/
WorkingFolder = "/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/"

# s3://vault-data-corpus/vessel data/H2O Generated Data/
OutputDir = "/Users/cv0361/Desktop/TechChallenge/Data/csv/H2O Generated Data/"
```

In [12]:
```
# Combining all Stats Data for All Years
df_list = list()

for FileName in glob(WorkingFolder + "Statistic*"):
    print(FileName)
    df = pd.read_csv(FileName, sep=",")

    print("Rows:", len(df))

    df_list.append(df)

#     break

Stat = pd.concat(df_list, ignore_index=True)
print("Total Rows:", Stat.shape)

# Remove duplicate vessel records after combining all the zones/years
Stat.drop_duplicates(inplace=True)

print("Non-Dup Total Rows:", Stat.shape)
# Vessel.reset_index(inplace=True)
```

```
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2009.csv
Rows: 1540
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2016.csv
Rows: 381
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2017.csv
Rows: 437
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2015.csv
Rows: 387
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2014.csv
Rows: 2078
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2010.csv
Rows: 1556
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2011.csv
Rows: 1755
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2013.csv
Rows: 1801
/Users/cv0361/Desktop/TechChallenge/Data/csv/ConsolidatedAIS/Statistic_2012.csv
Rows: 1725
Total Rows: (11660, 18)
Non-Dup Total Rows: (11660, 18)
```

In [13]:
```
Stat.head()
```

Out[13]:

	mmsi_id	PingRecStart	PingRecEnd	TotalPing	LatStd	LonStd	MaxSOG	MinSOG	MeanSOG	MedianSOG	StdSOG	MaxCOG	MinCOG	Mea
0	367047170	2008-12-31	2009-01-31	38078	0.184945	0.240594	102.0	0.0	3.384710	1.0	4.041475	360.0	0.0	179.5
1	366763770	2008-12-31	2009-01-31	27019	0.107984	0.110426	15.0	0.0	0.980051	0.0	2.397335	359.0	0.0	187.4
2	368494000	2008-12-31	2009-01-31	43422	0.000037	0.000035	0.0	0.0	0.000000	0.0	0.000000	360.0	0.0	180.0
3	366116000	2008-12-31	2009-01-09	11907	0.163482	0.227486	10.0	0.0	0.538339	0.0	1.993043	360.0	0.0	249.2
4	316003289	2008-12-31	2009-01-31	43378	0.195541	0.275708	18.0	0.0	6.057564	0.0	6.823938	360.0	0.0	175.0

In [14]:
```
essel = pd.read_csv(WorkingFolder + "Vessel.csv", sep=",")

essel['vessel_type'] = Vessel['vessel_type'].fillna(0)
essel = Vessel.astype({"vessel_type": int})           # cast type to int

 Unified Vessel Type
essel.loc[Vessel.vessel_type.isin([30, 1001, 1002]), 'vessel_type'] = 1001                              # Fishing
essel.loc[Vessel.vessel_type.isin([36,37,1019]), 'vessel_type'] = 1019                                  # Pleasure Cr
essel.loc[Vessel.vessel_type.isin([1012,1013,1014,1015]), 'vessel_type'] = 1012                         # Passenger
essel.loc[Vessel.vessel_type.isin([31, 32, 52, 1025]), 'vessel_type'] = 1025                            # Tug Tow
essel.loc[Vessel.vessel_type.isin([80,81,82,83,84,85,86,87,88,89,1017, 1024]), 'vessel_type'] = 1024    # Tanker
essel.loc[Vessel.vessel_type.isin([70,71,72,73,74,75,76,77,78,79,1003,1004,1016]), 'vessel_type'] = 1016  # Cargo

essel.head()
```

Out[14]:

	mmsi_id	imo	call_sign	vessel_name	vessel_type	length	width	TempId	Id_len	StartDigit
0	303159000	IMO8315724	WAP2210	ARCTURUS	1001	39.93	9.76	303159000	9	3
1	367011410	IMO8856510	WAJ6882	KUSTATAN	1001	26.76	8.53	367011410	9	3
2	366499000	IMO6931055	WASF	KATIE ANN	1001	89.92	13.52	366499000	9	3
3	367528690	IMO7742358	WDG3692	ALASKAN LADY	1001	51.08	9.76	367528690	9	3
4	371542000	IMO8714944	3FSS6	NO1 POHAH	1016	115.00	16.00	371542000	9	3

In [15]:
```
# Vessel.vessel_type.drop_duplicates().sort_values().tail(20)
```

In [16]:
```
Vessel = Vessel.loc[Vessel.vessel_type.isin([1001,36,1012,1025,1024,1016])]
Vessel.shape
```

Out[16]: (6965, 10)

In [17]:
```
df = Vessel.merge(Stat, left_on="mmsi_id", right_on="mmsi_id", how="inner")
df.head()
```

Out[17]:

	mmsi_id	imo	call_sign	vessel_name	vessel_type	length	width	TempId	Id_len	StartDigit	...	MeanSOG	MedianSOG	StdSOG	Max
0	303159000	IMO8315724	WAP2210	ARCTURUS	1001	39.93	9.76	303159000	9	3	...	0.531859	0.0	2.259193	3
1	303159000	IMO8315724	WAP2210	ARCTURUS	1001	39.93	9.76	303159000	9	3	...	2.151764	0.0	3.698764	2
2	303159000	IMO8315724	WAP2210	ARCTURUS	1001	39.93	9.76	303159000	9	3	...	0.000976	0.0	0.009834	2
3	303159000	IMO8315724	WAP2210	ARCTURUS	1001	39.93	9.76	303159000	9	3	...	0.805481	0.0	2.469827	2
4	367011410	IMO8856510	WAJ6882	KUSTATAN	1001	26.76	8.53	367011410	9	3	...	3.353605	3.1	2.959427	2

5 rows × 27 columns

In [18]:
```
df = df[['vessel_type', 'length', 'width', 'PingRecStart', 'PingRecEnd',
       'TotalPing', 'LatStd', 'LonStd', 'MaxSOG', 'MinSOG', 'MeanSOG',
       'MedianSOG', 'StdSOG', 'MaxCOG', 'MinCOG', 'MeanCOG', 'MedianCOG',
       'StdCOG', 'AnoThreshold', 'AnoClusterCount']]

df = df.drop_duplicates()
df.reset_index(inplace=True)
df.head()
```

Out[18]:

	index	vessel_type	length	width	PingRecStart	PingRecEnd	TotalPing	LatStd	LonStd	MaxSOG	...	MeanSOG	MedianSOG	StdSOG	MaxCOG
0	0	1001	39.93	9.76	2008-12-31	2009-01-15	20622	0.155418	0.426351	14.0	...	0.531859	0.0	2.259193	360.0
1	1	1001	39.93	9.76	2016-01-18	2016-01-30	5158	0.245250	0.313110	11.2	...	2.151764	0.0	3.698764	204.7
2	2	1001	39.93	9.76	2017-01-30	2017-01-31	1537	0.000008	0.000011	0.1	...	0.000976	0.0	0.009834	204.7
3	3	1001	39.93	9.76	2015-01-01	2015-01-31	30102	0.587081	0.801748	11.4	...	0.805481	0.0	2.469827	204.7
4	4	1001	26.76	8.53	2016-01-10	2016-01-30	9807	1.553294	2.342727	12.1	...	3.353605	3.1	2.959427	204.7

5 rows × 21 columns

In [19]:
```
# Output Stat Data
df.to_csv(OutputDir + "AIS_Classify_VesselType.csv", index=None, header = True)
```