Anomaly detection using h20.ai

This notebook uses tools from h20.ai to classify suspicious (anomalous) activity from AIS vessel data.

In [12]:
```
import numpy as np
import pandas as pd
import xgboost as xgb
import gc
import random
import csv
import math
from sklearn import metrics
import h2o
from h2o.automl import H2OAutoML


random.seed(24)
np.random.seed(seed=24)
set_label = 2000

# s3://vault-data-corpus/vessel data/H2O Generated Data/
main = '/Users/kimmontgomery/Documents/POC_DOD/DOD4/'
h2o.init()
```

```
Checking whether there is an H2O instance running at http://localhost:54321 (http://localhost:54321) . connecte
d.
```

H2O_cluster_uptime:	3 hours 38 mins
H2O_cluster_timezone:	America/Denver
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.33.0.5216
H2O_cluster_version_age:	2 months and 25 days
H2O_cluster_name:	H2O_from_python_kimmontgomery_obdaca
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	2.942 Gb
H2O_cluster_total_cores:	12
H2O_cluster_allowed_cores:	12
H2O_cluster_status:	locked, healthy
H2O_connection_url:	http://localhost:54321
H2O_connection_proxy:	{"http": null, "https": null}
H2O_internal_security:	False
H2O_API_Extensions:	Amazon S3, XGBoost, Algos, AutoML, Core V3, TargetEncoder, Core V4
Python_version:	3.7.7 final

In [13]:

```
train = h2o.import_file(main + "train_with_features.csv")
test = h2o.import_file(main + "test_with_features.csv")

#train['target']= train.apply(lambda row: int((row['distance_from_predicted'] > 10.0) and (row['distance_from_prev

# If the columns corresponding to the point being predicted are still present, drop them
drop_columns =  ['X_5', 'Y_5', 'Z_5', 'SOG_5', 'COG_5', 'Heading_5', 'ROT_5', 'BaseDateTime_5',
                 'Status_5', 'VoyageID_5', 'MMSI_5', 'ReceiverType_5', 'ReceiverID_5', 'dataset_5',
                 'time_gap_5', 'predicted_x', 'predicted_y']

# Drop these columns if still present
drop_columns += ['distance_from_predicted', 'distance_from_previous']
drop_columns += ['Z_1', 'Z_2', 'Z_3', 'Z_4']
drop_columns += [ 'dataset_2', 'dataset_3','dataset_4']
drop_columns += ['BaseDateTime_1', 'BaseDateTime_2',  'BaseDateTime_3',  'BaseDateTime_4']

train_columns = list(train.columns)

keep_columns = [item for item in train_columns if item not in drop_columns]

# Identify predictors and response
x = keep_columns
y = "target"
x.remove(y)

# For binary classification, response should be a factor
train[y] = train[y].asfactor()
test[y] = test[y].asfactor()
```

```
Parse progress: |████████████████████████████████████████████████████████| 100%
Parse progress: |████████████████████████████████████████████████████████| 100%
```

In [14]:
```
# Run AutoML for up to an hour
aml = H2OAutoML(max_runtime_secs=3600, seed=1)
aml.train(x=x, y=y, training_frame=train)

# View the AutoML Leaderboard
lb = aml.leaderboard
lb.head(rows=lb.nrows)  # Print all rows instead of default (10 rows)
```

```
AutoML progress: |████████████████████████████████████████████████████████| 100%
```

model_id	auc	logloss	aucpr	mean_per_class_error	rmse	mse
StackedEnsemble_BestOfFamily_AutoML_20210102_223301	0.809324	0.177293	0.26174	0.34006	0.213802	0.0457112
StackedEnsemble_AllModels_AutoML_20210102_223301	0.809271	0.177735	0.257077	0.348198	0.214216	0.0458885
GBM_grid__1_AutoML_20210102_223301_model_3	0.79976	0.175524	0.23995	0.328774	0.213855	0.045734
GBM_grid__1_AutoML_20210102_223301_model_5	0.798104	0.177718	0.250494	0.342229	0.213564	0.0456096
GBM_5_AutoML_20210102_223301	0.797257	0.177034	0.229998	0.346558	0.214711	0.0461007
XGBoost_grid__1_AutoML_20210102_223301_model_5	0.797253	0.178279	0.23262	0.350224	0.215399	0.0463968
GBM_4_AutoML_20210102_223301	0.79485	0.181075	0.232912	0.350313	0.21558	0.0464747
GBM_grid__1_AutoML_20210102_223301_model_10	0.793978	0.177283	0.241694	0.367089	0.21412	0.0458476
XGBoost_grid__1_AutoML_20210102_223301_model_28	0.793922	0.176663	0.231948	0.352608	0.214384	0.0459604
GBM_2_AutoML_20210102_223301	0.793805	0.178737	0.237917	0.358027	0.215069	0.0462546
XGBoost_grid__1_AutoML_20210102_223301_model_16	0.793434	0.176654	0.23313	0.358035	0.214187	0.045876
GBM_grid__1_AutoML_20210102_223301_model_7	0.793271	0.177584	0.243418	0.377769	0.214216	0.0458885
GBM_grid__1_AutoML_20210102_223301_model_8	0.793162	0.177935	0.244948	0.357527	0.214118	0.0458465
GBM_grid__1_AutoML_20210102_223301_model_6	0.791091	0.182078	0.235542	0.369535	0.215598	0.0464826
XGBoost_grid__1_AutoML_20210102_223301_model_31	0.790705	0.203509	0.244763	0.332189	0.217089	0.0471276
XRT_1_AutoML_20210102_223301	0.788402	0.189956	0.241086	0.348676	0.213975	0.0457854
GBM_grid__1_AutoML_20210102_223301_model_4	0.788202	0.18001	0.229832	0.344244	0.215297	0.0463528
XGBoost_grid__1_AutoML_20210102_223301_model_44	0.787809	0.181318	0.224405	0.327903	0.216282	0.046778
GBM_grid__1_AutoML_20210102_223301_model_9	0.787558	0.182466	0.225419	0.351245	0.217046	0.0471088
XGBoost_grid__1_AutoML_20210102_223301_model_14	0.78645	0.180929	0.223708	0.359807	0.216618	0.0469235
XGBoost_grid__1_AutoML_20210102_223301_model_25	0.786327	0.186538	0.225813	0.352559	0.217778	0.0474273
XGBoost_grid__1_AutoML_20210102_223301_model_7	0.786303	0.207925	0.214201	0.350957	0.220077	0.0484338
DRF_1_AutoML_20210102_223301	0.78571	0.186791	0.242453	0.361104	0.214013	0.0458017
GBM_3_AutoML_20210102_223301	0.785467	0.180622	0.237337	0.335529	0.215493	0.0464372
XGBoost_grid__1_AutoML_20210102_223301_model_27	0.785375	0.18002	0.230012	0.386647	0.215524	0.0464507
XGBoost_grid__1_AutoML_20210102_223301_model_10	0.784995	0.199074	0.211195	0.318175	0.219801	0.0483126
XGBoost_grid__1_AutoML_20210102_223301_model_47	0.784926	0.196994	0.221976	0.356868	0.218081	0.0475594
XGBoost_grid__1_AutoML_20210102_223301_model_26	0.783827	0.180893	0.224711	0.324938	0.216293	0.0467827
XGBoost_grid__1_AutoML_20210102_223301_model_48	0.783416	0.178584	0.227568	0.351326	0.214786	0.046133
GBM_1_AutoML_20210102_223301	0.783413	0.185988	0.211806	0.361251	0.220272	0.0485199
XGBoost_grid__1_AutoML_20210102_223301_model_37	0.783296	0.181127	0.229073	0.358355	0.216086	0.0466931
XGBoost_grid__1_AutoML_20210102_223301_model_18	0.782706	0.182142	0.224991	0.369397	0.216594	0.0469131
XGBoost_grid__1_AutoML_20210102_223301_model_29	0.78268	0.178506	0.230965	0.354546	0.214636	0.0460687
XGBoost_grid__1_AutoML_20210102_223301_model_43	0.781047	0.18426	0.208199	0.368157	0.217683	0.0473861
XGBoost_grid__1_AutoML_20210102_223301_model_32	0.781002	0.187092	0.206888	0.339296	0.218935	0.0479323
XGBoost_grid__1_AutoML_20210102_223301_model_33	0.780825	0.187308	0.213684	0.380096	0.218985	0.0479545
XGBoost_3_AutoML_20210102_223301	0.780615	0.185882	0.208107	0.345688	0.218593	0.0477831
XGBoost_grid__1_AutoML_20210102_223301_model_23	0.779626	0.181945	0.212946	0.336172	0.216593	0.0469124
XGBoost_grid__1_AutoML_20210102_223301_model_38	0.779287	0.180287	0.212624	0.349455	0.215831	0.0465831
GBM_grid__1_AutoML_20210102_223301_model_1	0.779185	0.179449	0.23701	0.358863	0.214254	0.0459046
XGBoost_grid__1_AutoML_20210102_223301_model_20	0.779077	0.232167	0.207463	0.362203	0.222525	0.0495174
XGBoost_grid__1_AutoML_20210102_223301_model_40	0.778653	0.179487	0.228078	0.366377	0.214977	0.046215
XGBoost_grid__1_AutoML_20210102_223301_model_15	0.778527	0.189051	0.20481	0.358367	0.218854	0.0478971

model_id	auc	logloss	aucpr	mean_per_class_error	rmse	mse
XGBoost_grid__1_AutoML_20210102_223301_model_24	0.778366	0.180503	0.216162	0.354962	0.215865	0.0465976
XGBoost_grid__1_AutoML_20210102_223301_model_17	0.778235	0.184153	0.209442	0.370664	0.217846	0.0474567
XGBoost_grid__1_AutoML_20210102_223301_model_41	0.778098	0.224857	0.220331	0.383866	0.220621	0.0486736
GBM_grid__1_AutoML_20210102_223301_model_2	0.777684	0.183988	0.222016	0.346805	0.217035	0.0471042
XGBoost_grid__1_AutoML_20210102_223301_model_46	0.777081	0.18329	0.216115	0.366362	0.216987	0.0470836
XGBoost_grid__1_AutoML_20210102_223301_model_19	0.775147	0.195709	0.225135	0.349894	0.21823	0.0476244
XGBoost_grid__1_AutoML_20210102_223301_model_39	0.77472	0.183347	0.206869	0.348489	0.217764	0.0474213
XGBoost_grid__1_AutoML_20210102_223301_model_42	0.774234	0.18483	0.219991	0.371172	0.216875	0.0470349
XGBoost_grid__1_AutoML_20210102_223301_model_34	0.774023	0.184589	0.216358	0.376514	0.217098	0.0471317
XGBoost_grid__1_AutoML_20210102_223301_model_6	0.773697	0.195271	0.214723	0.352906	0.220382	0.048568
XGBoost_grid__1_AutoML_20210102_223301_model_22	0.771443	0.184132	0.226386	0.347895	0.216749	0.0469799
XGBoost_grid__1_AutoML_20210102_223301_model_35	0.771222	0.185745	0.210537	0.355263	0.218038	0.0475405
XGBoost_grid__1_AutoML_20210102_223301_model_12	0.770401	0.199873	0.199868	0.343452	0.220743	0.0487273
XGBoost_grid__1_AutoML_20210102_223301_model_21	0.770339	0.192273	0.222235	0.383119	0.2176	0.04735
XGBoost_grid__1_AutoML_20210102_223301_model_1	0.76997	0.227197	0.208003	0.352036	0.221627	0.0491186
XGBoost_grid__1_AutoML_20210102_223301_model_11	0.769747	0.190463	0.197065	0.353577	0.220083	0.0484367
XGBoost_2_AutoML_20210102_223301	0.769078	0.187438	0.198892	0.384802	0.218874	0.0479056
XGBoost_grid__1_AutoML_20210102_223301_model_2	0.767551	0.206187	0.193327	0.334552	0.223752	0.0500647
XGBoost_grid__1_AutoML_20210102_223301_model_36	0.767168	0.193248	0.207406	0.39064	0.219233	0.0480631
XGBoost_grid__1_AutoML_20210102_223301_model_13	0.76597	0.189982	0.205401	0.376958	0.219771	0.0482994
XGBoost_grid__1_AutoML_20210102_223301_model_45	0.765176	0.18735	0.201113	0.357428	0.218443	0.0477175
XGBoost_grid__1_AutoML_20210102_223301_model_9	0.761818	0.187127	0.209623	0.349637	0.218593	0.047783
XGBoost_grid__1_AutoML_20210102_223301_model_8	0.760486	0.18996	0.200737	0.361148	0.219396	0.0481347
XGBoost_grid__1_AutoML_20210102_223301_model_4	0.75933	0.235901	0.18402	0.375286	0.224996	0.0506233
XGBoost_grid__1_AutoML_20210102_223301_model_3	0.75834	0.192108	0.198971	0.360132	0.220061	0.0484267
XGBoost_grid__1_AutoML_20210102_223301_model_30	0.756666	0.202664	0.202392	0.385268	0.220683	0.048701
XGBoost_1_AutoML_20210102_223301	0.751576	0.197762	0.194716	0.352305	0.221193	0.0489263
DeepLearning_grid__2_AutoML_20210102_223301_model_3	0.722868	0.261569	0.146564	0.377719	0.233776	0.0546511
DeepLearning_grid__3_AutoML_20210102_223301_model_1	0.711799	0.209647	0.147966	0.376509	0.221028	0.0488533
DeepLearning_grid__2_AutoML_20210102_223301_model_1	0.708925	0.220845	0.150471	0.378661	0.222398	0.049461
DeepLearning_grid__3_AutoML_20210102_223301_model_3	0.700817	0.230244	0.125648	0.367665	0.224079	0.0502115
DeepLearning_grid__1_AutoML_20210102_223301_model_1	0.696493	0.25434	0.141177	0.366412	0.225798	0.0509848
DeepLearning_grid__1_AutoML_20210102_223301_model_3	0.677375	0.321899	0.127745	0.387358	0.23894	0.0570925
DeepLearning_grid__3_AutoML_20210102_223301_model_2	0.677015	0.355945	0.122135	0.371421	0.230948	0.0533372
GLM_1_AutoML_20210102_223301	0.671486	0.199986	0.128856	0.3897	0.222516	0.0495134
DeepLearning_grid__1_AutoML_20210102_223301_model_4	0.656487	0.401805	0.100488	0.387888	0.25343	0.0642268
DeepLearning_grid__3_AutoML_20210102_223301_model_4	0.653372	0.336736	0.109213	0.403986	0.230043	0.0529199
DeepLearning_grid__1_AutoML_20210102_223301_model_2	0.652932	0.368928	0.112009	0.377005	0.232598	0.0541018
DeepLearning_grid__2_AutoML_20210102_223301_model_2	0.651672	0.367839	0.104222	0.406178	0.23038	0.0530748
DeepLearning_1_AutoML_20210102_223301	0.650489	0.222775	0.0976176	0.400837	0.22923	0.0525466
DeepLearning_grid__2_AutoML_20210102_223301_model_4	0.647163	0.358645	0.0984093	0.398475	0.231626	0.0536505

Out[14]:

```
In [15]: exa = aml.explain(test)
```

Leaderboard

> Leaderboard shows models with their metrics. When provided with H2OAutoML object, the leaderboard shows 5-fold cross-validated metrics by default (depending on the H2OAutoML settings), otherwise it shows metrics computed on the frame. At most 20 models are shown by default.

model_id	auc	logloss	aucpr	mean_per_class_error	rmse	mse	training_time_ms	predic
StackedEnsemble_BestOfFamily_AutoML_20210102_223301	0.809324	0.177293	0.26174	0.34006	0.213802	0.0457112	603	
StackedEnsemble_AllModels_AutoML_20210102_223301	0.809271	0.177735	0.257077	0.348198	0.214216	0.0458885	1579	
GBM_grid__1_AutoML_20210102_223301_model_3	0.79976	0.175524	0.23995	0.328774	0.213855	0.045734	950	
GBM_grid__1_AutoML_20210102_223301_model_5	0.798104	0.177718	0.250494	0.342229	0.213564	0.0456096	1116	
GBM_5_AutoML_20210102_223301	0.797257	0.177034	0.229998	0.346558	0.214711	0.0461007	1101	
XGBoost_grid__1_AutoML_20210102_223301_model_5	0.797253	0.178279	0.23262	0.350224	0.215399	0.0463968	1640	
GBM_4_AutoML_20210102_223301	0.79485	0.181075	0.232912	0.350313	0.21558	0.0464747	1014	
GBM_grid__1_AutoML_20210102_223301_model_10	0.793978	0.177283	0.241694	0.367089	0.21412	0.0458476	608	
XGBoost_grid__1_AutoML_20210102_223301_model_28	0.793922	0.176663	0.231948	0.352608	0.214384	0.0459604	2727	
GBM_2_AutoML_20210102_223301	0.793805	0.178737	0.237917	0.358027	0.215069	0.0462546	688	
XGBoost_grid__1_AutoML_20210102_223301_model_16	0.793434	0.176654	0.23313	0.358035	0.214187	0.045876	5741	
GBM_grid__1_AutoML_20210102_223301_model_7	0.793271	0.177584	0.243418	0.377769	0.214216	0.0458885	532	
GBM_grid__1_AutoML_20210102_223301_model_8	0.793162	0.177935	0.244948	0.357527	0.214118	0.0458465	650	
GBM_grid__1_AutoML_20210102_223301_model_6	0.791091	0.182078	0.235542	0.369535	0.215598	0.0464826	925	
XGBoost_grid__1_AutoML_20210102_223301_model_31	0.790705	0.203509	0.244763	0.332189	0.217089	0.0471276	2516	
XRT_1_AutoML_20210102_223301	0.788402	0.189956	0.241086	0.348676	0.213975	0.0457854	1536	
GBM_grid__1_AutoML_20210102_223301_model_4	0.788202	0.18001	0.229832	0.344244	0.215297	0.0463528	1186	
XGBoost_grid__1_AutoML_20210102_223301_model_44	0.787809	0.181318	0.224405	0.327903	0.216282	0.046778	2535	
GBM_grid__1_AutoML_20210102_223301_model_9	0.787558	0.182466	0.225419	0.351245	0.217046	0.0471088	620	
XGBoost_grid__1_AutoML_20210102_223301_model_14	0.78645	0.180929	0.223708	0.359807	0.216618	0.0469235	2363	

Confusion Matrix

> Confusion matrix shows a predicted class vs an actual class.

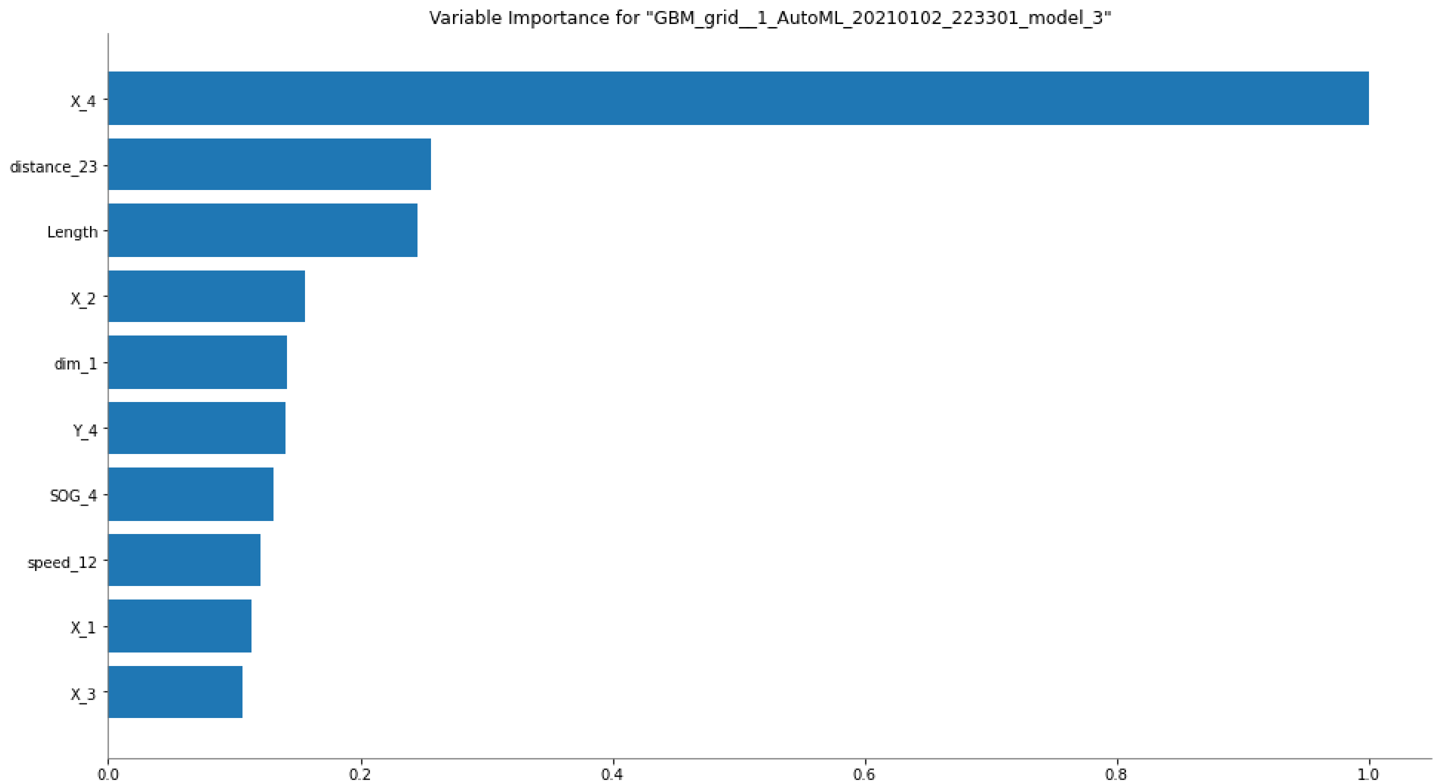
StackedEnsemble_BestOfFamily_AutoML_20210102_223301

```
Confusion Matrix (Act/Pred) for max f1 @ threshold = 0.2198812898218783:
```

		0	1	Error	Rate
0	0	8091.0	22.0	0.0027	(22.0/8113.0)
1	1	17.0	445.0	0.0368	(17.0/462.0)
2	Total	8108.0	467.0	0.0045	(39.0/8575.0)

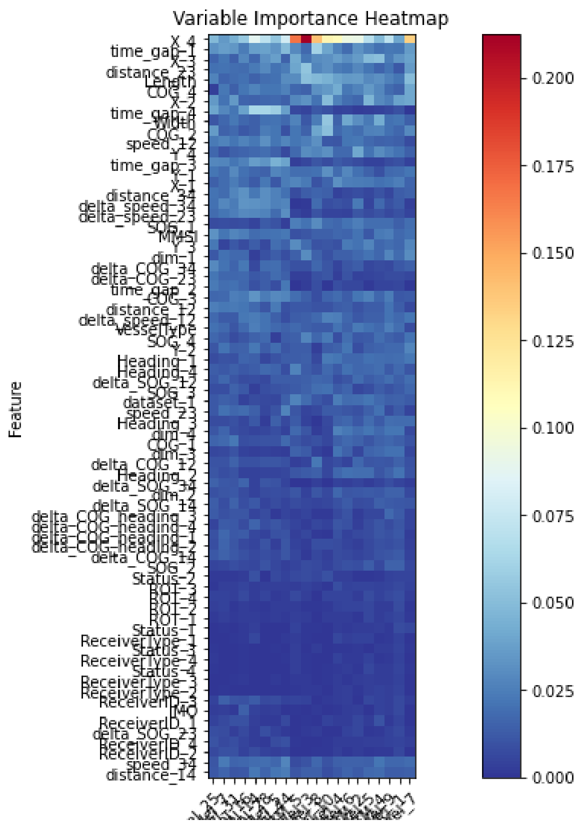
Variable Importance

> The variable importance plot shows the relative importance of the most important variables in the model.



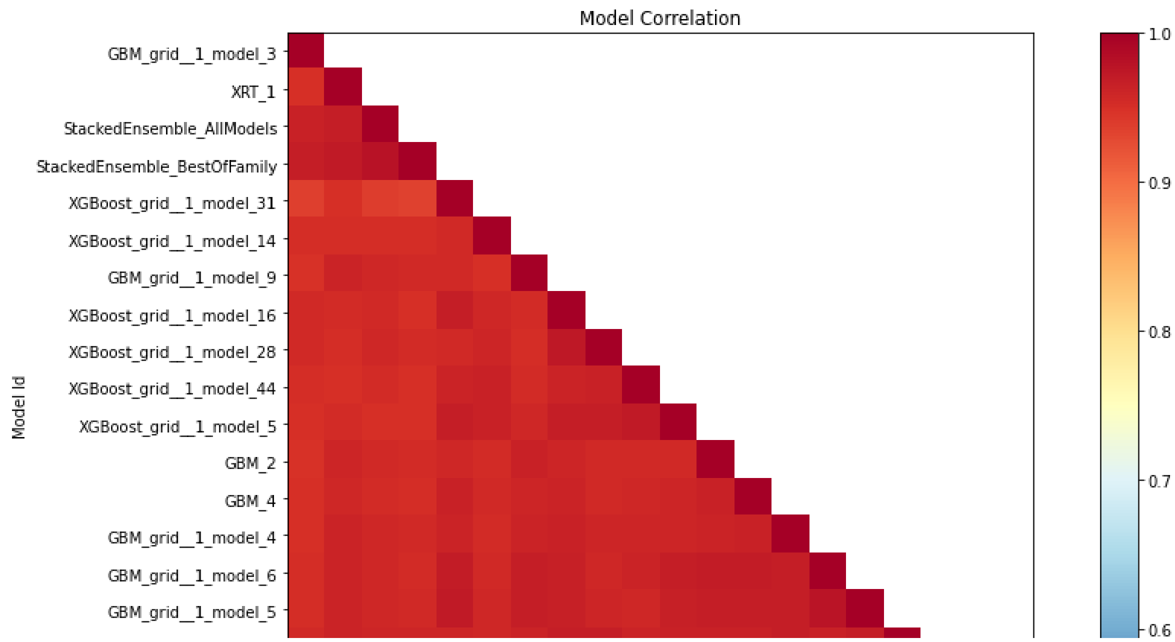
Variable Importance Heatmap

> Variable importance heatmap shows variable importance across multiple models. Some models in H2O return variable importance for one-hot (binary indicator) encoded versions of categorical columns (e.g. Deep Learning, XGBoost). In order for the variable importance of categorical columns to be compared across all model types we compute a summarization of the the variable importance across all one-hot encoded features and return a single variable importance for the original categorical feature. By default, the models and variables are ordered by their similarity.



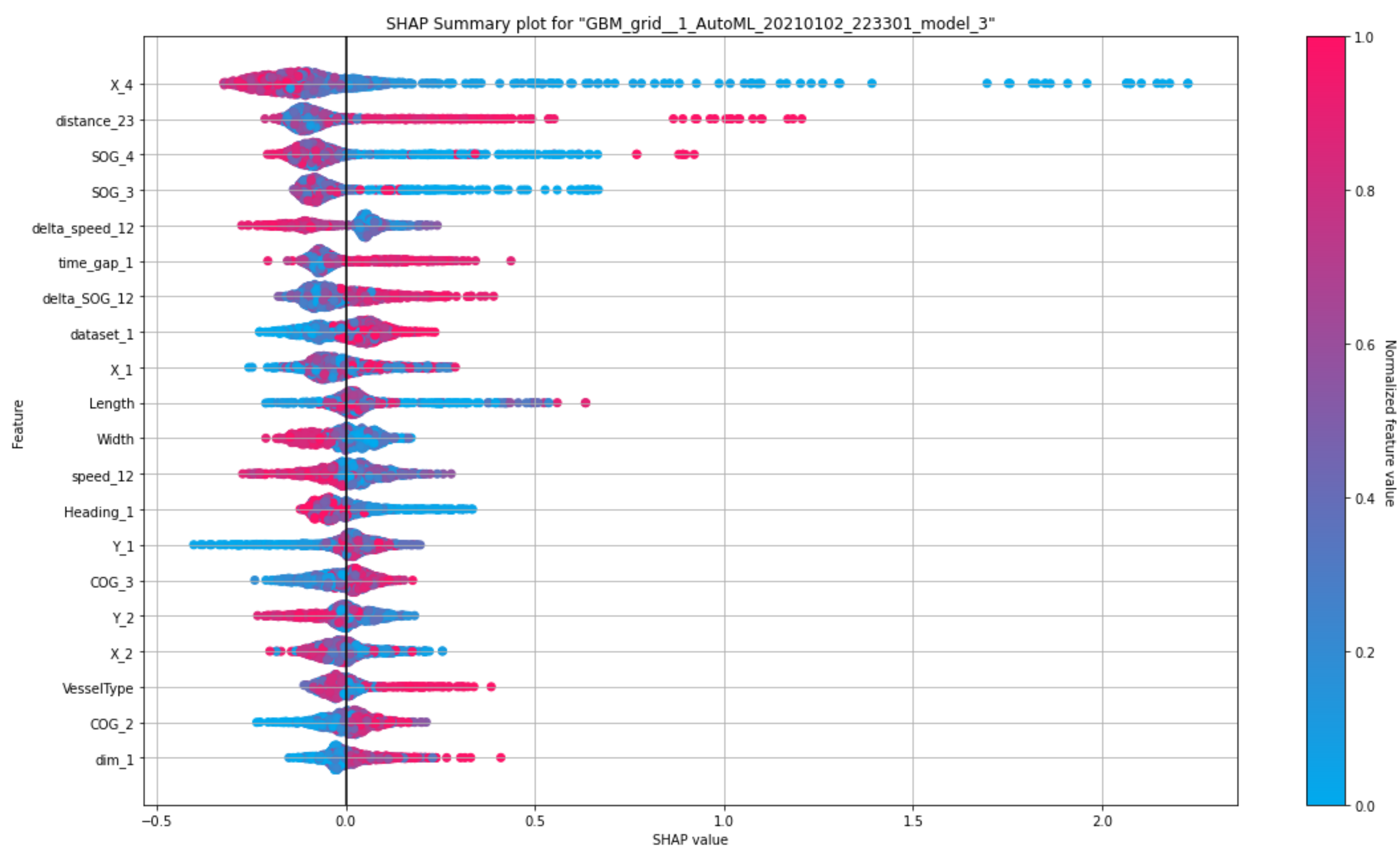
Model Correlation

This plot shows the correlation between the predictions of the models. For classification, frequency of identical predictions is used. By default, models are ordered by their similarity (as computed by hierarchical clustering). Interpretable models, such as GAM, GLM, and RuleFit are highlighted using red colored text.



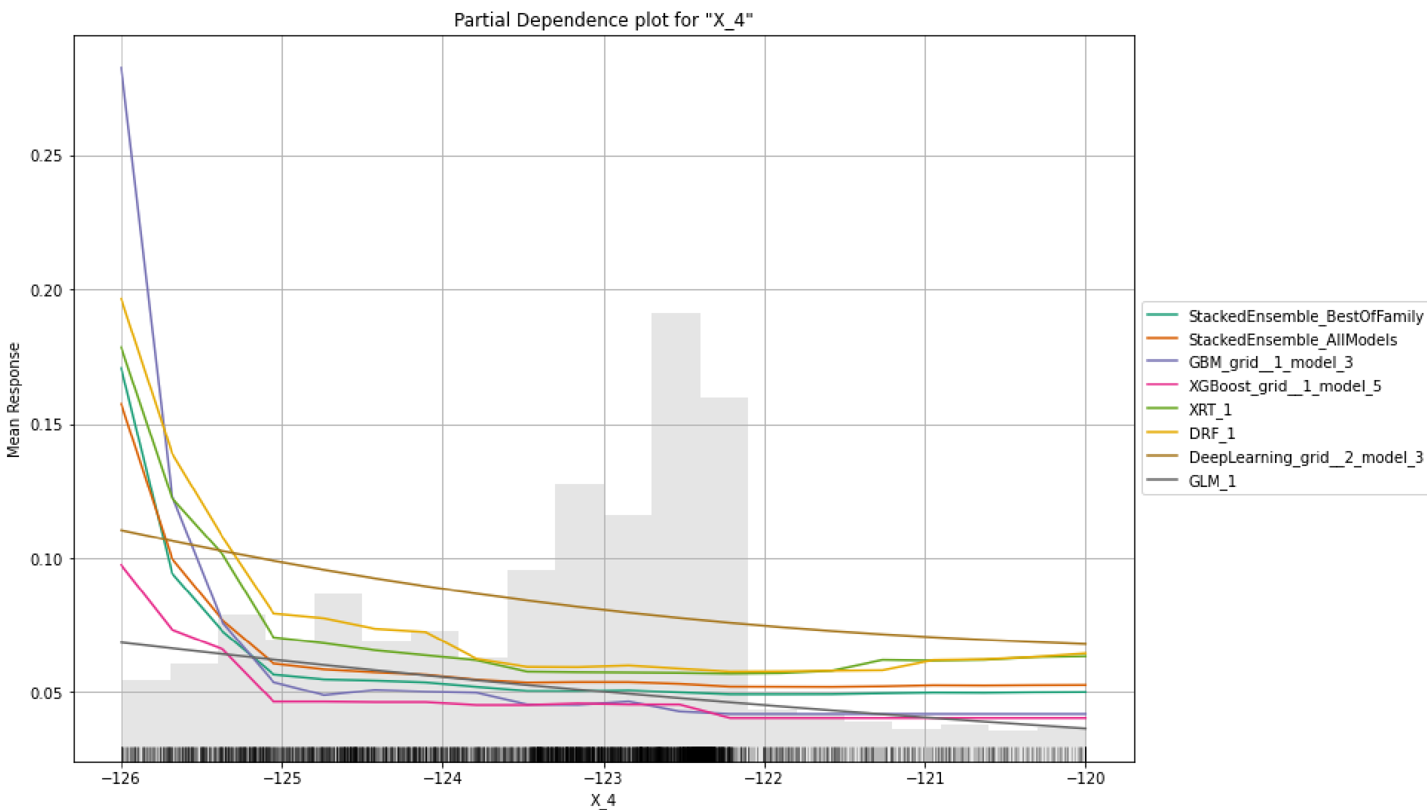
SHAP Summary

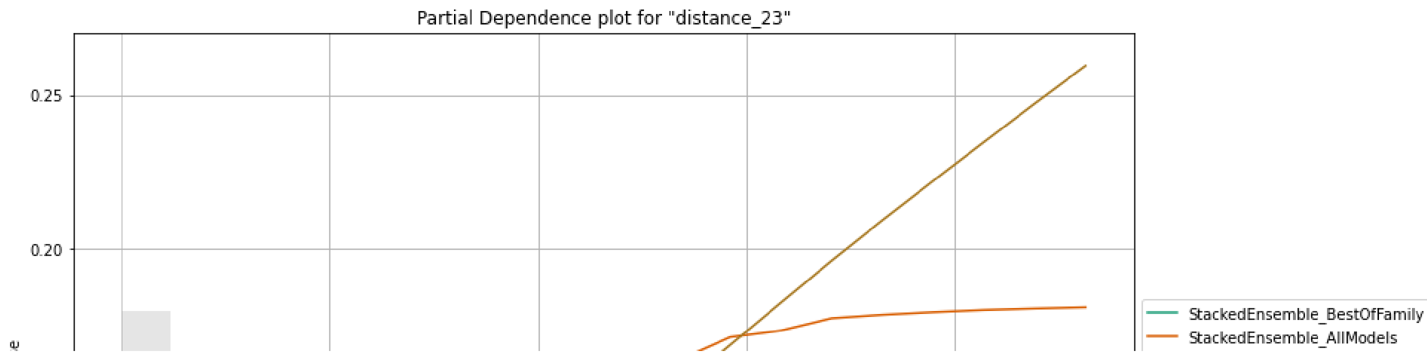
> SHAP summary plot shows the contribution of the features for each instance (row of data). The sum of the feature contributions and the bias term is equal to the raw prediction of the model, i.e., prediction before applying inverse link function.

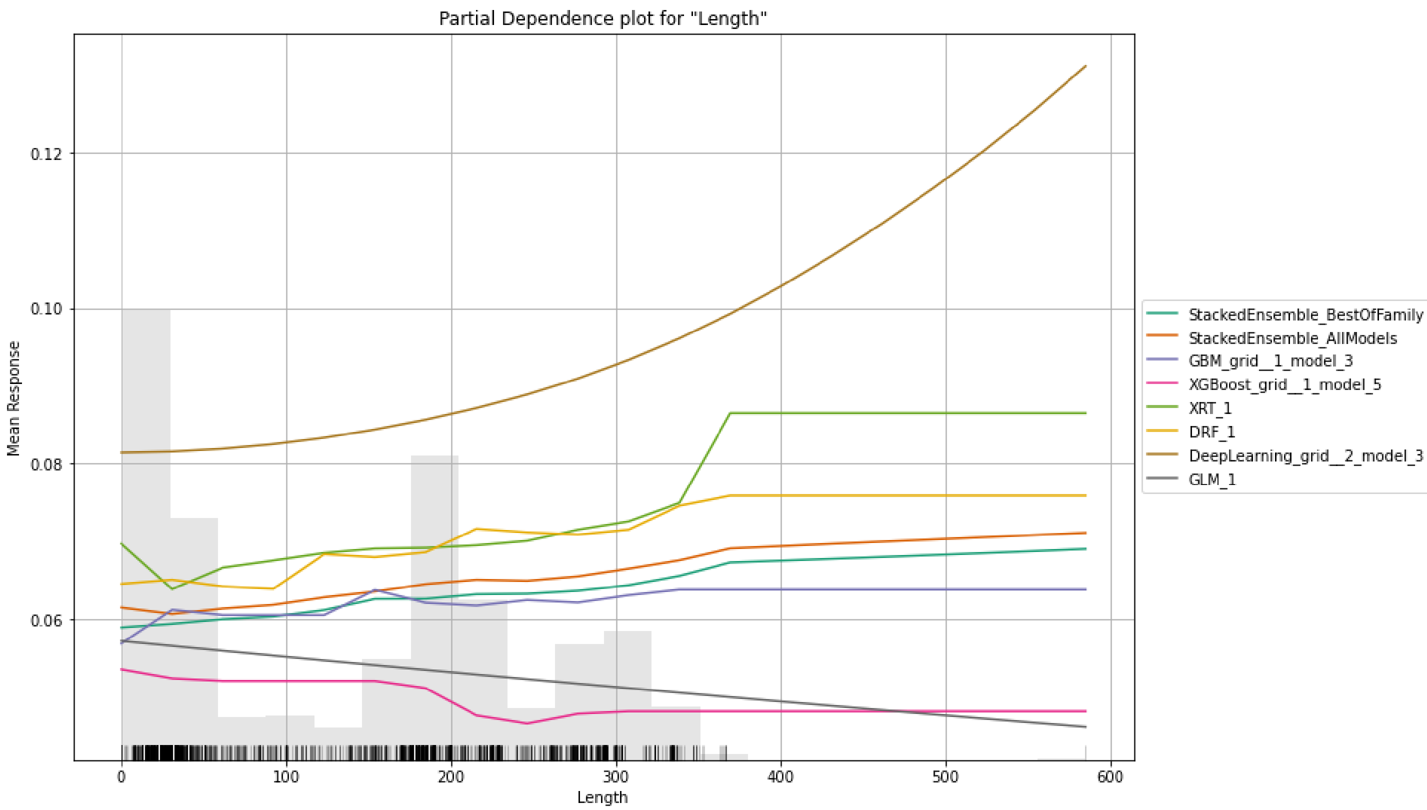


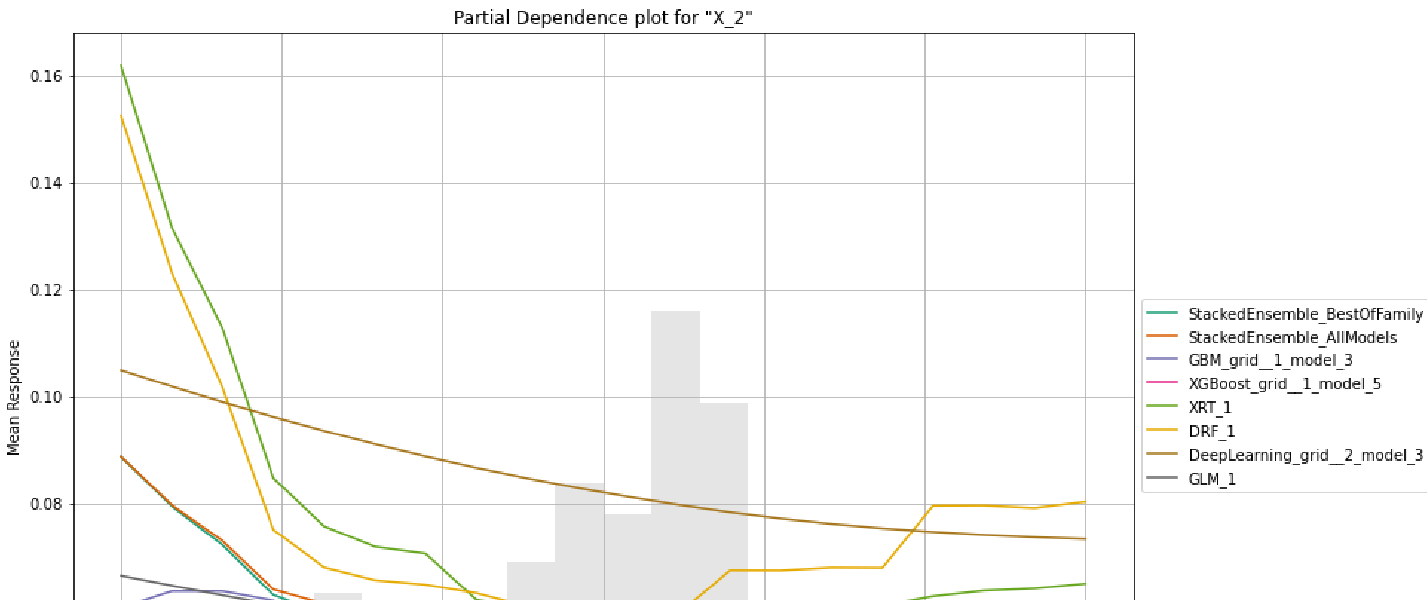
Partial Dependence Plots

> Partial dependence plot (PDP) gives a graphical depiction of the marginal effect of a variable on the response. The effect of a variable is measured in change in the mean response. PDP assumes independence between the feature for which is the PDP computed and the rest.

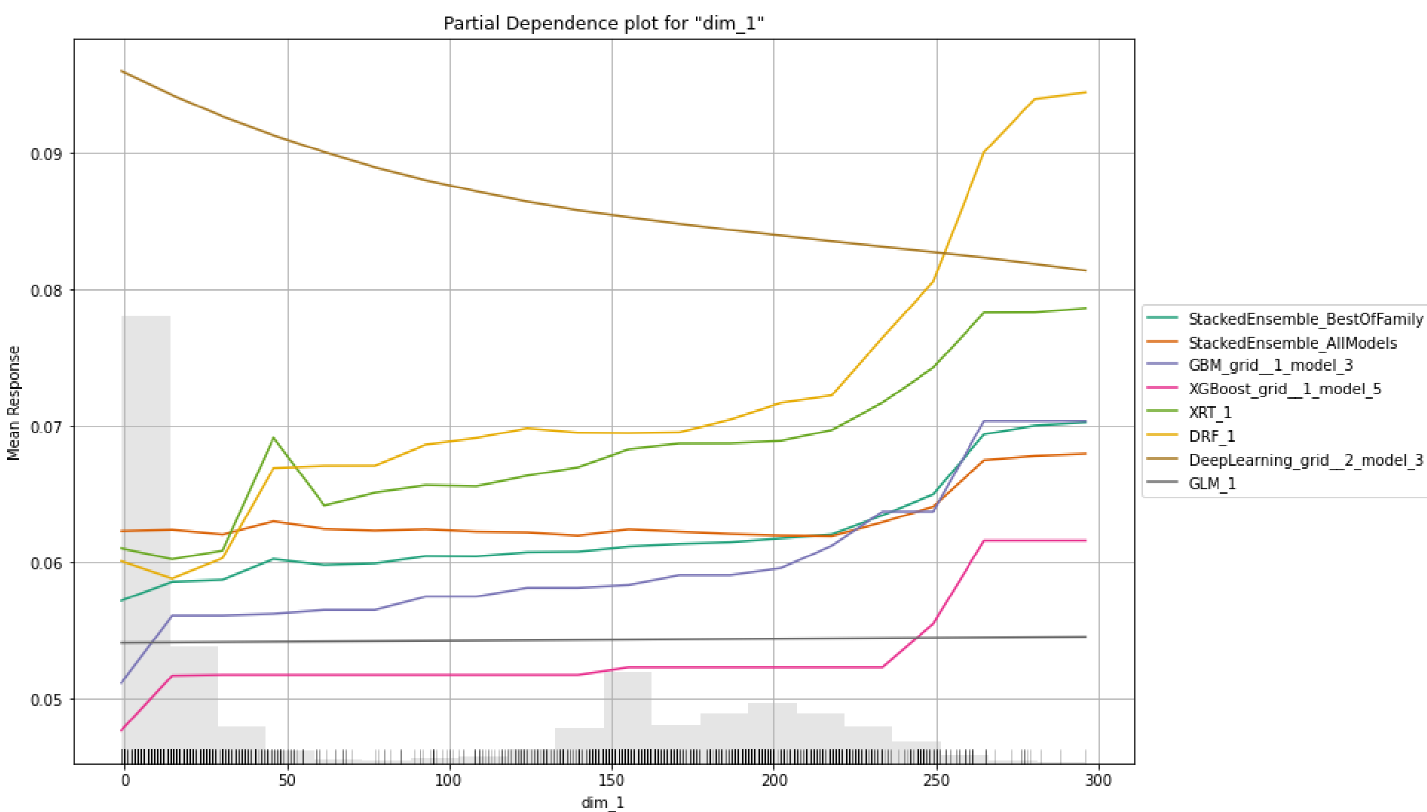



Partial Dependence plot for "distance_23"
0.25
0.20
StackedEnsemble_BestOfFamily
StackedEnsemble_AllModels


Partial Dependence plot for "Length"
0.12
0.10
0.08
0.06
Mean Response
0
100
200
300
400
500
600
Length
StackedEnsemble_BestOfFamily
StackedEnsemble_AllModels
GBM_grid__1_model_3
XGBoost_grid__1_model_5
XRT_1
DRF_1
DeepLearning_grid__2_model_3
GLM_1



Partial Dependence plot for "X_2"
Mean Response
0.16
0.14
0.12
0.10
0.08
StackedEnsemble_BestOfFamily
StackedEnsemble_AllModels
GBM_grid__1_model_3
XGBoost_grid__1_model_5
XRT_1
DRF_1
DeepLearning_grid__2_model_3
GLM_1



Partial Dependence plot for "dim_1"
Mean Response
0.09
0.08
0.07
0.06
0.05
0
50
100
150
200
250
300
dim_1
StackedEnsemble_BestOfFamily
StackedEnsemble_AllModels
GBM_grid__1_model_3
XGBoost_grid__1_model_5
XRT_1
DRF_1
DeepLearning_grid__2_model_3
GLM_1