

```
In [10]: import h2o
import numpy as np #to efficiently compute functions
import pandas as pd #to work with data frames
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
import zipfile #to open zipfile
pd.set_option('display.max_columns', 35) #to display 35 columns
pd.set_option('display.max_rows', 100) #to display 100 rows
from ydata_profiling import ProfileReport
import os

# data partition
from sklearn.model_selection import train_test_split

#filter methods
# spearman
# chi-square
import scipy.stats as stats
from scipy.stats import pearsonr, spearmanr
from scipy.stats import chi2_contingency

#wrapper methods
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.feature_selection import RFE
from sklearn.preprocessing import LabelEncoder

# embedded methods
from sklearn.linear_model import LassoCV

import warnings
warnings.filterwarnings('ignore')
```

```
In [4]: h2o.init()
```

```
Checking whether there is an H2O instance running at http://localhost:54321.....
not found.
Attempting to start a local H2O server...
; Java HotSpot(TM) 64-Bit Server VM (build 25.381-b09, mixed mode)
Starting server from C:\Users\alexg\AppData\Local\Programs\Python\Python38\Lib
\site-packages\h2o\backend\bin\h2o.jar
Ice root: C:\Users\alexg\AppData\Local\Temp\tmp2ugbzazh
JVM stdout: C:\Users\alexg\AppData\Local\Temp\tmp2ugbzazh\h2o_alexg_started_fro
m_python.out
JVM stderr: C:\Users\alexg\AppData\Local\Temp\tmp2ugbzazh\h2o_alexg_started_fro
m_python.err
Server is running at http://127.0.0.1:54321
Connecting to H2O server at http://127.0.0.1:54321 ... successful.
```

H2O_cluster_uptime:	04 secs
H2O_cluster_timezone:	Europe/London
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.46.0.6
H2O_cluster_version_age:	4 days
H2O_cluster_name:	H2O_from_python_alexg_d68ck2
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	3.507 Gb
H2O_cluster_total_cores:	12
H2O_cluster_allowed_cores:	12
H2O_cluster_status:	locked, healthy
H2O_connection_url:	http://127.0.0.1:54321
H2O_connection_proxy:	{"http": null, "https": null}
H2O_internal_security:	False
Python_version:	3.8.5 final

In [5]: `from h2o.automl import H2OAutoML`

```
In [11]: train_zip_path = 'train_data.csv.zip'
test_zip_path = 'test_data.csv.zip'

# open csv file from zip
def read_zipped_csv(zip_path):
    with zipfile.ZipFile(zip_path, 'r') as z:
        csv_filename = z.namelist()[0]

        with z.open(csv_filename) as f:
            return pd.read_csv(f)

train_df = read_zipped_csv(train_zip_path)
test_df = read_zipped_csv(test_zip_path)
"""
print("Train Data:")
print(train_df.head())
print("Test Data:")
print(test_df.head()) """
```

Out[11]: ' \nprint("Train Data:")\nprint(train_df.head())\nprint("Test Data:")\nprint(test_df.head()) '

In [15]: `train_df.columns`

```
Out[15]: Index(['Accident Date', 'Age at Injury', 'Alternative Dispute Resolution',
               'Assembly Date', 'Attorney/Representative', 'Average Weekly Wage',
               'Birth Year', 'C-2 Date', 'C-3 Date', 'Carrier Name', 'Carrier Type',
               'Claim Identifier', 'Claim Injury Type', 'County of Injury',
               'COVID-19 Indicator', 'District Name', 'First Hearing Date', 'Gender',
               'IME-4 Count', 'Industry Code', 'Industry Code Description',
               'Medical Fee Region', 'OIICS Nature of Injury Description',
               'WCIO Cause of Injury Code', 'WCIO Cause of Injury Description',
               'WCIO Nature of Injury Code', 'WCIO Nature of Injury Description',
               'WCIO Part Of Body Code', 'WCIO Part Of Body Description', 'Zip Code',
               'Agreement Reached', 'WCB Decision', 'Number of Dependents'],
              dtype='object')
```

```
In [16]: X_train , X_test, y_train, y_test = train_test_split(train_df.drop(columns=['Clas']), y_train, test_size=0.2, random_state=42)
```

```
In [20]: X_train = X_train.drop('Claim Identifier', axis=1)
```

```
In [21]: aml = H2OAutoML(max_models=10, seed = 10, exclude_algos=['StackedEnsemble', 'De
```

```
In [24]: # Combine X train and y train into a single DataFrame
```

```
train df = pd.concat([X train, y train], axis=1)
```

```
test_df = pd.concat([X_test, y_test], axis=1)
```

```
# Convert pandas DataFrames to H2OFrames
```

```
training frame = h2o.H2OFrame(train df)
```

```
validation_frame = h2o.H2OFrame(test_df)
```

```
# Define the target and features
```

```
x = X_train.columns.tolist()
```

```
y = 'Claim Injury Type'
```

[illegible]

```
Parse progress: | ██████████  
| (done) 100%
```

```
In [25]: # Train the model
```

```
aml.train(x=x, y=y, training frame=training frame, validation frame=validation f
```

AutoML progress: |
16:04:06.26: Project: AutoML_1_20241106_160406
16:04:06.26: Cross-validation disabled by user: no fold column nor nfolds > 1.
16:04:06.27: Setting stopping tolerance adaptively based on the training frame:
0.001451294740596166
16:04:06.27: Build control seed: 10
16:04:06.27: training frame: Frame key: AutoML_1_20241106_160406_training_Key_Fra
me__upload_b91c2cbfc004af32ce6bb22963e241ff.hex cols: 32 rows: 474776 chun
ks: 35 size: 34202826 checksum: 2738228822042061110
16:04:06.27: validation frame: Frame key: Key_Frame__upload_87c4ccf5cb8b29346b30e
126b91d4075.hex cols: 33 rows: 118695 chunks: 9 size: 9171928 checksu
m: 5683256690309732508
16:04:06.27: leaderboard frame: Frame key: Key_Frame__upload_87c4ccf5cb8b29346b30
e126b91d4075.hex cols: 33 rows: 118695 chunks: 9 size: 9171928 checksu
m: 5683256690309732508
16:04:06.27: blending frame: NULL
16:04:06.27: response column: Claim Injury Type
16:04:06.27: fold column: null
16:04:06.27: weights column: null
16:04:06.34: AutoML: XGBoost is not available; skipping it.
16:04:06.38: Loading execution steps: [{XGBoost : [def_2 (1g, 10w), def_1 (2g, 10
w), def_3 (3g, 10w), grid_1 (4g, 90w), lr_search (7g, 30w)]}, {GLM : [def_1 (1g,
10w)]}, {DRF : [def_1 (2g, 10w), XRT (3g, 10w)]}, {GBM : [def_5 (1g, 10w), def_2
(2g, 10w), def_3 (2g, 10w), def_4 (2g, 10w), def_1 (3g, 10w), grid_1 (4g, 60w), l
r_annealing (7g, 10w)]}, {DeepLearning : [def_1 (3g, 10w), grid_1 (4g, 30w), grid
_2 (5g, 30w), grid_3 (5g, 30w)]}, {completion : [resume_best_grids (6g, 60w)]},
{StackedEnsemble : [monotonic (9g, 10w), best_of_family_xglm (10g, 10w), all_xglm
(10g, 10w)]}]
16:04:06.58: Disabling Algo: DeepLearning as requested by the user.
16:04:06.58: Disabling Algo: XGBoost as requested by the user.
16:04:06.58: Disabling Algo: StackedEnsemble as requested by the user.
16:04:06.58: AutoML job created: 2024.11.06 16:04:06.0
16:04:06.58: AutoML build started: 2024.11.06 16:04:06.58
16:04:06.94: AutoML: starting GLM_1_AutoML_1_20241106_160406 model training
16:04:06.129: _train param, Dropping bad and constant columns: [OIICS Nature of I
njury Description]

16:21:44.822: New leader: GLM_1_AutoML_1_20241106_160406, mean_per_class_error:
0.6316641020704011
16:21:44.832: AutoML: starting GBM_1_AutoML_1_20241106_160406 model training
16:21:44.833: _train param, Dropping bad and constant columns: [OIICS Nature of I
njury Description]
16:25:09.303: New leader: GBM_1_AutoML_1_20241106_160406, mean_per_class_error:
0.5120992584684063
16:25:09.304: AutoML: starting DRF_1_AutoML_1_20241106_160406 model training
16:25:09.304: _train param, Dropping bad and constant columns: [OIICS Nature of I
njury Description]

16:28:10.343: AutoML: starting GBM_2_AutoML_1_20241106_160406 model training
16:28:10.343: _train param, Dropping bad and constant columns: [OIICS Nature of I
njury Description]
16:30:06.102: New leader: GBM_2_AutoML_1_20241106_160406, mean_per_class_error:
0.5072393793990744
16:30:06.102: AutoML: starting GBM_3_AutoML_1_20241106_160406 model training
16:30:06.102: _train param, Dropping bad and constant columns: [OIICS Nature of I
njury Description]

16:32:25.849: AutoML: starting GBM_4_AutoML_1_20241106_160406 model training
16:32:25.849: _train param, Dropping bad and constant columns: [OIICS Nature of Injury Description]

16:35:12.605: AutoML: starting XRT_1_AutoML_1_20241106_160406 model training
16:35:12.606: _train param, Dropping bad and constant columns: [OIICS Nature of Injury Description]

16:37:35.955: AutoML: starting GBM_5_AutoML_1_20241106_160406 model training
16:37:35.955: _train param, Dropping bad and constant columns: [OIICS Nature of Injury Description]

16:40:00.170: New leader: GBM_5_AutoML_1_20241106_160406, mean_per_class_error: 0.5046975750789643
16:40:00.172: AutoML: starting GBM_grid_1_AutoML_1_20241106_160406 hyperparameter search

16:41:25.148: AutoML: starting GBM_grid_1_AutoML_1_20241106_160406 hyperparameter search

█ (done) 100%

16:50:30.197: Skipping StackedEnsemble 'monotonic' due to the exclude_algos option or it is already trained.
16:50:30.197: Skipping StackedEnsemble 'best_of_family_xglm' due to the exclude_algos option or it is already trained.
16:50:30.197: Skipping StackedEnsemble 'all_xglm' due to the exclude_algos option or it is already trained.
16:50:30.197: Actual modeling steps: [{GLM : [def_1 (1g, 10w)]}, {GBM : [def_5 (1g, 10w)]}, {DRF : [def_1 (2g, 10w)]}, {GBM : [def_2 (2g, 10w), def_3 (2g, 10w), def_4 (2g, 10w)]}, {DRF : [XRT (3g, 10w)]}, {GBM : [def_1 (3g, 10w), grid_1 (4g, 60w)]}, {completion : [resume_best_grids (6g, 60w)]}]
16:50:30.197: AutoML build stopped: 2024.11.06 16:50:30.197
16:50:30.197: AutoML build done: built 13 models
16:50:30.197: AutoML duration: 46 min 24.139 sec

Out[25]:

Model Details

=====

H2OGradientBoostingEstimator : Gradient Boosting Machine

Model Key: GBM_5_AutoML_1_20241106_160406

Model Summary:

number_of_trees	number_of_internal_trees	model_size_in_bytes	min_depth	max_depth
135.0	1215.0	1264052.0	6.0	

ModelMetricsMultinomial: gbm

** Reported on train data. **

MSE: 0.1322185092503756

RMSE: 0.363618631605114

LogLoss: 0.4238102659093432

Mean Per-Class Error: 0.2598751178810194

AUC table was not computed: it is either disabled (model parameter 'auc_type' was set to AUTO or NONE) or the domain size exceeds the limit (maximum is 50 domains).

AUCPR table was not computed: it is either disabled (model parameter 'auc_type' was set to AUTO or NONE) or the domain size exceeds the limit (maximum is 50 domains).

Confusion Matrix: Row labels: Actual class; Column labels: Predicted class

	1. CANCELLED	2. NON-COMP	3. MED ONLY	4. TEMPORARY	5. PPD SCH LOSS	6. PPD NSL	7. PTD	8. DEATH	nan
	6127.0	3577.0	214.0	76.0	20.0	0.0	0.0	0.0	0.0
	1272.0	228997.0	1333.0	808.0	144.0	1.0	0.0	1.0	0.0
	68.0	28217.0	8310.0	16119.0	2507.0	7.0	0.0	1.0	0.0
	9.0	1751.0	1191.0	109903.0	6062.0	17.0	0.0	7.0	0.0
	5.0	75.0	389.0	8762.0	29321.0	12.0	0.0	0.0	0.0
	0.0	2.0	33.0	1870.0	473.0	986.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	76.0	0.0	0.0
	1.0	4.0	1.0	18.0	0.0	0.0	0.0	355.0	0.0

1. CANCELLED	2. NON- COMP	3. MED ONLY	4. TEMPORARY	5. PPD SCH LOSS	6. PPD NSL	7. PTD	8. DEATH	nan
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15654.0
7482.0	262623.0	11471.0	137556.0	38527.0	1023.0	76.0	364.0	15654.0

Top-9 Hit Ratios:

k	hit_ratio
1	0.8419318
2	0.9739203
3	0.9955958
4	0.9993745
5	0.9999242
6	1.0
7	1.0
8	1.0
9	1.0

ModelMetricsMultinomial: gbm
** Reported on validation data. **

MSE: 0.15231184176749946
RMSE: 0.39027149750846457
LogLoss: 0.495271589957532
Mean Per-Class Error: 0.5046975750789643
AUC table was not computed: it is either disabled (model parameter 'auc_type' was set to AUTO or NONE) or the domain size exceeds the limit (maximum is 50 domains).
AUCPR table was not computed: it is either disabled (model parameter 'auc_type' was set to AUTO or NONE) or the domain size exceeds the limit (maximum is 50 domains).

Confusion Matrix: Row labels: Actual class; Column labels: Predicted class

1. CANCELLED	2. NON- COMP	3. MED ONLY	4. TEMPORARY	5. PPD SCH LOSS	6. PPD NSL	7. PTD	8. DEATH	nan
1242.0	1140.0	59.0	19.0	3.0	0.0	0.0	0.0	0.0

1. CANCELLED	2. NON- COMP	3. MED ONLY	4. TEMPORARY	5. PPD SCH LOSS	6. PPD NSL	7. PTD	8. DEATH	nan	
421.0	57355.0	486.0	215.0	35.0	2.0	0.0	8.0	0.0	0.019
20.0	7323.0	1431.0	4211.0	676.0	11.0	0.0	5.0	0.0	0.895
6.0	484.0	463.0	26618.0	1945.0	39.0	4.0	8.0	0.0	0.099
2.0	26.0	132.0	2845.0	6702.0	9.0	0.0	0.0	0.0	0.310
0.0	3.0	10.0	656.0	156.0	22.0	0.0	0.0	0.0	0.974
0.0	0.0	0.0	21.0	0.0	0.0	0.0	0.0	0.0	
0.0	16.0	4.0	48.0	0.0	0.0	0.0	23.0	0.0	0.747
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3791.0	
1691.0	66347.0	2585.0	34633.0	9517.0	83.0	4.0	44.0	3791.0	0.181

Top-9 Hit Ratios:

k	hit_ratio
1	0.8187708
2	0.9635031
3	0.9920216
4	0.9983403
5	0.9993429
6	0.9997051
7	0.9998568
8	1.0
9	1.0

Scoring History:

timestamp	duration	number_of_trees	training_rmse	training_logloss	training_cla
-----------	----------	-----------------	---------------	------------------	--------------

timestamp	duration	number_of_trees	training_rmse	training_logloss	training_classification
2024-11-06 16:37:35	0.030 sec	0.0	0.8888889	2.1972246	
2024-11-06 16:37:41	5.242 sec	5.0	0.6886348	1.2525880	
2024-11-06 16:37:46	10.593 sec	10.0	0.5907160	0.9838916	
2024-11-06 16:37:54	18.621 sec	15.0	0.5430302	0.8610535	
2024-11-06 16:38:00	24.057 sec	20.0	0.4936928	0.7257665	
2024-11-06 16:38:05	29.438 sec	25.0	0.4442698	0.6136519	
2024-11-06 16:38:10	34.689 sec	30.0	0.4263784	0.5726548	
2024-11-06 16:38:16	40.111 sec	35.0	0.4115376	0.5393458	
2024-11-06 16:38:21	45.446 sec	40.0	0.4024984	0.5178863	
2024-11-06 16:38:26	50.899 sec	45.0	0.3961978	0.5027511	
-					
-	---	---	---	---	---
-					
2024-11-06 16:39:14	1 min 38.980 sec	90.0	0.3732852	0.4466142	
2024-11-06 16:39:20	1 min 44.444 sec	95.0	0.3720592	0.4435395	
2024-11-06 16:39:25	1 min 49.758 sec	100.0	0.3703238	0.4393938	
2024-11-06 16:39:31	1 min 55.134 sec	105.0	0.3690045	0.4363293	
2024-11-06 16:39:36	2 min 0.150 sec	110.0	0.3679320	0.4337405	
2024-11-06 16:39:41	2 min 5.205 sec	115.0	0.3669912	0.4315724	
2024-11-06 16:39:46	2 min 10.046 sec	120.0	0.3661741	0.4296801	

timestamp	duration	number_of_trees	training_rmse	training_logloss	training_classification
2024-11-06 16:39:50	2 min 14.655 sec	125.0	0.3652722	0.4276521	
2024-11-06 16:39:55	2 min 19.212 sec	130.0	0.3645782	0.4259808	
2024-11-06 16:39:59	2 min 23.829 sec	135.0	0.3636186	0.4238103	

[28 rows x 14 columns]

Variable Importances:

variable	relative_importance	scaled_importance	percentage
IME-4 Count	127514.4921875	1.0	0.1827648
Average Weekly Wage	122359.6562500	0.9595745	0.1753765
Attorney/Representative	122206.4765625	0.9583732	0.1751569
Agreement Reached	71971.6640625	0.5644195	0.1031560
WCIO Nature of Injury Description	37942.0039062	0.2975505	0.0543818
First Hearing Date	36402.8085938	0.2854798	0.0521757
County of Injury	30755.8867188	0.2411952	0.0440820
WCIO Part Of Body Description	25992.6328125	0.2038406	0.0372549
WCIO Cause of Injury Description	20675.9296875	0.1621457	0.0296345
C-3 Date	15971.9550781	0.1252560	0.0228924
---	---	---	---
Number of Dependents	2338.9614258	0.0183427	0.0033524
C-2 Date	1654.6660156	0.0129763	0.0023716
Gender	1617.9577637	0.0126884	0.0023190
Age at Injury	1134.5015869	0.0088970	0.0016261
WCIO Nature of Injury Code	935.6208496	0.0073374	0.0013410
WCIO Part Of Body Code	359.5622559	0.0028198	0.0005154
Zip Code	301.6259155	0.0023654	0.0004323
Birth Year	245.1838379	0.0019228	0.0003514
WCB Decision	242.4812012	0.0019016	0.0003475
WCIO Cause of Injury Code	177.8856506	0.0013950	0.0002550

[30 rows x 4 columns]

```
[tips]
Use `model.explain()` to inspect the model.
--
Use `h2o.display.toggle_user_tips()` to switch on/off this section.
```

```
In [31]: aml.explain(validation_frame)
```

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	mean_per_class_error	logloss	rmse
	GBM_5_AutoML_1_20241106_160406	0.504698	0.495272	0.390271
	GBM_grid_1_AutoML_1_20241106_160406_model_7	0.50676	0.507204	0.392911
	GBM_2_AutoML_1_20241106_160406	0.507239	0.497173	0.389987
	GBM_3_AutoML_1_20241106_160406	0.511054	0.495761	0.389271
	GBM_1_AutoML_1_20241106_160406	0.512099	0.511056	0.391883
	GBM_grid_1_AutoML_1_20241106_160406_model_6	0.513743	0.517517	0.399297
	GBM_grid_1_AutoML_1_20241106_160406_model_4	0.514676	0.520395	0.402797
	GBM_grid_1_AutoML_1_20241106_160406_model_8	0.51763	0.515638	0.398954
	GBM_4_AutoML_1_20241106_160406	0.51946	0.503172	0.389322
	GBM_grid_1_AutoML_1_20241106_160406_model_1	0.531463	0.534073	0.408822
	DRF_1_AutoML_1_20241106_160406	0.581008	0.680965	0.471558
	XRT_1_AutoML_1_20241106_160406	0.614665	0.837479	0.526299
	GLM_1_AutoML_1_20241106_160406	0.631664	0.795053	0.516833

[13 rows x 8 columns]



Confusion Matrix

Confusion matrix shows a predicted class vs an actual class.

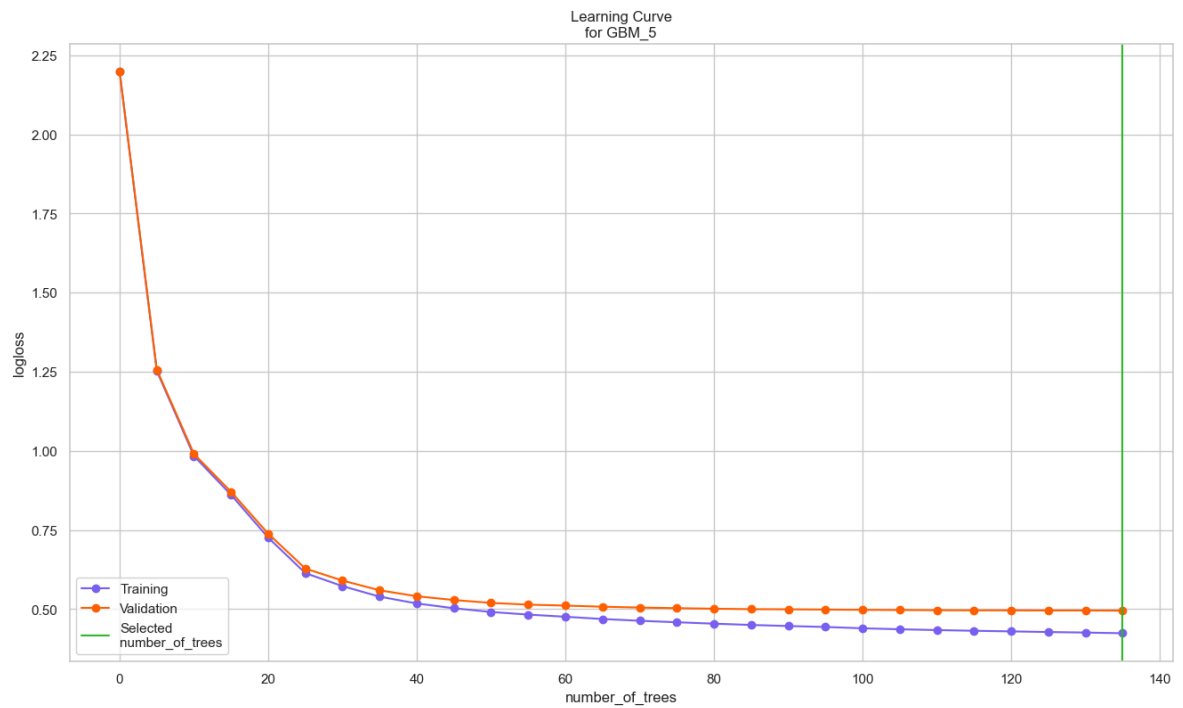
GBM_5_AutoML_1_20241106_160406

Confusion Matrix: Row labels: Actual class; Column labels: Predicted class

	1. CANCELLED	2. NON-COMP	3. MED ONLY	4. TEMPORARY	5. PPD SCH LOSS	6. PPD NSL	7. PTD	8. DEATH	nan	Err
	1242.0	1140.0	59.0	19.0	3.0	0.0	0.0	0.0	0.0	0.49573
	421.0	57355.0	486.0	215.0	35.0	2.0	0.0	8.0	0.0	0.01994
	20.0	7323.0	1431.0	4211.0	676.0	11.0	0.0	5.0	0.0	0.89537
	6.0	484.0	463.0	26618.0	1945.0	39.0	4.0	8.0	0.0	0.09973
	2.0	26.0	132.0	2845.0	6702.0	9.0	0.0	0.0	0.0	0.31021
	0.0	3.0	10.0	656.0	156.0	22.0	0.0	0.0	0.0	0.97402
	0.0	0.0	0.0	21.0	0.0	0.0	0.0	0.0	0.0	
	0.0	16.0	4.0	48.0	0.0	0.0	0.0	23.0	0.0	0.74725
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3791.0	
	1691.0	66347.0	2585.0	34633.0	9517.0	83.0	4.0	44.0	3791.0	0.18122

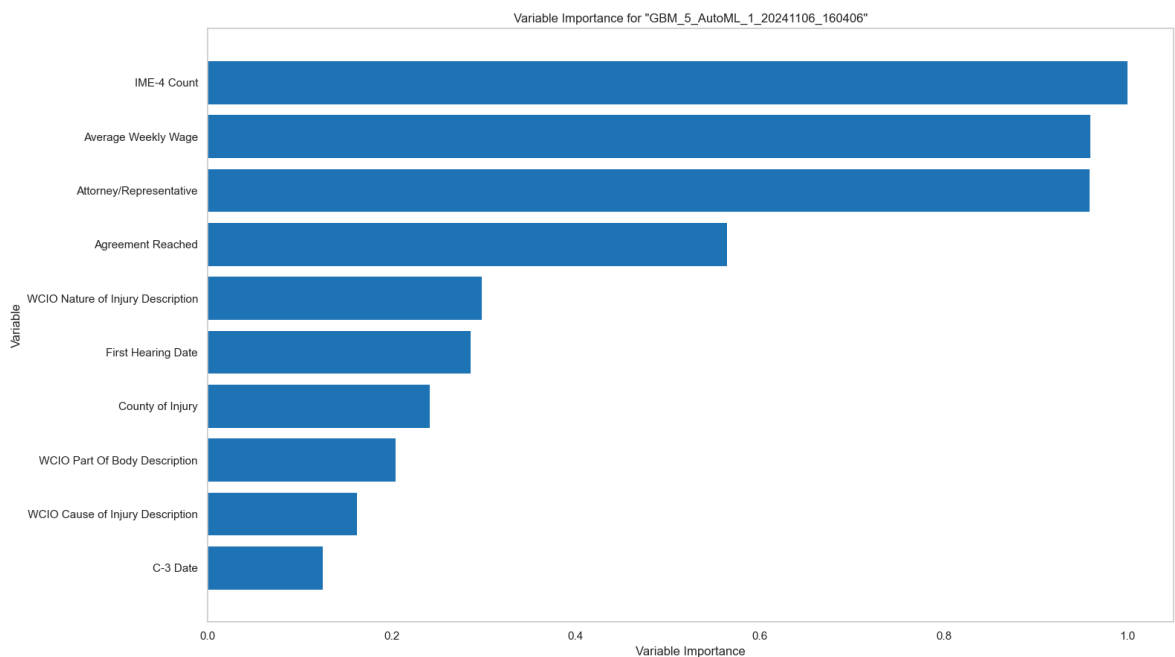
Learning Curve Plot

Learning curve plot shows the loss function/metric dependent on number of iterations or trees for tree-based algorithms. This plot can be useful for determining whether the model overfits.



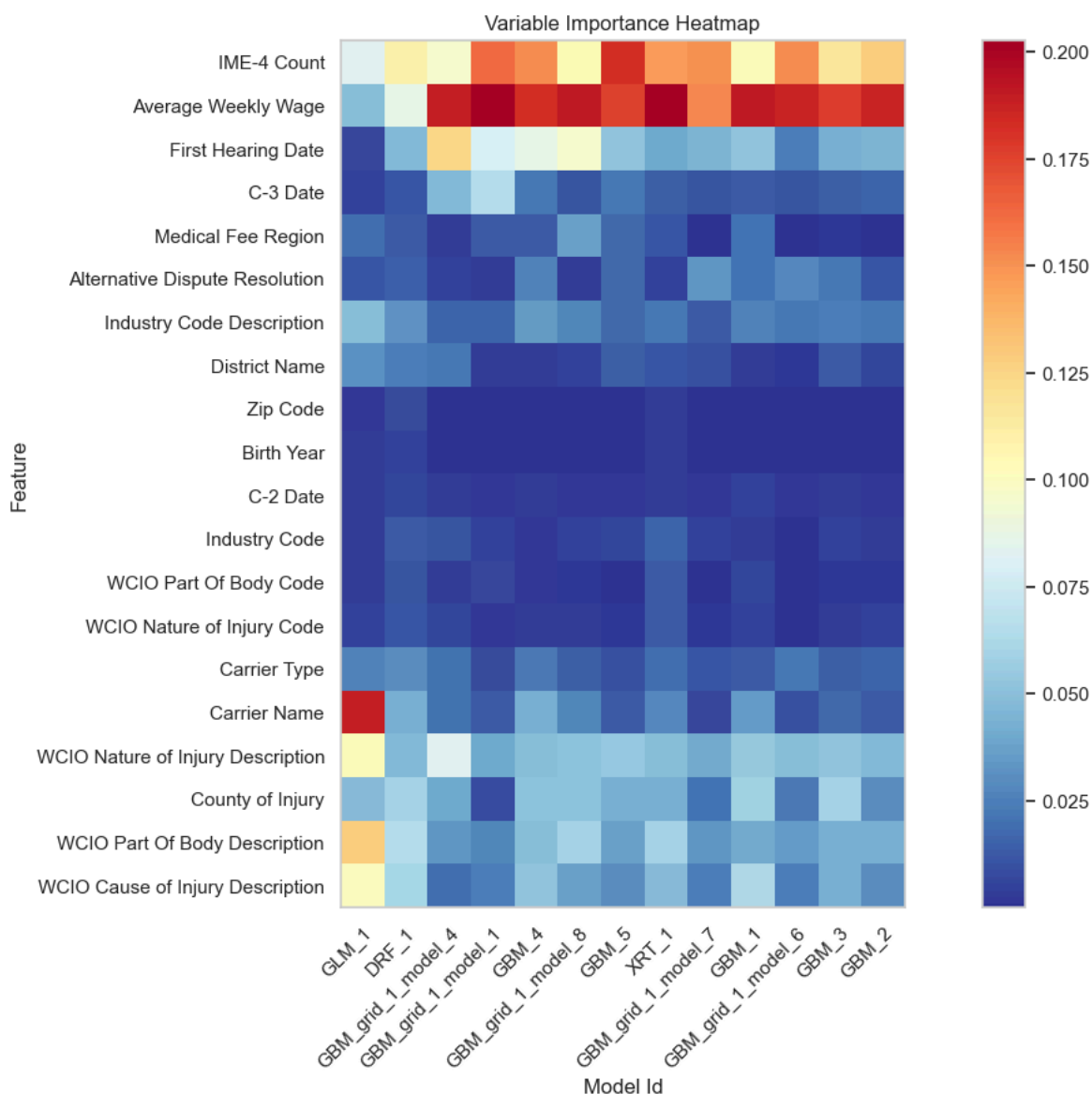
Variable Importance

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



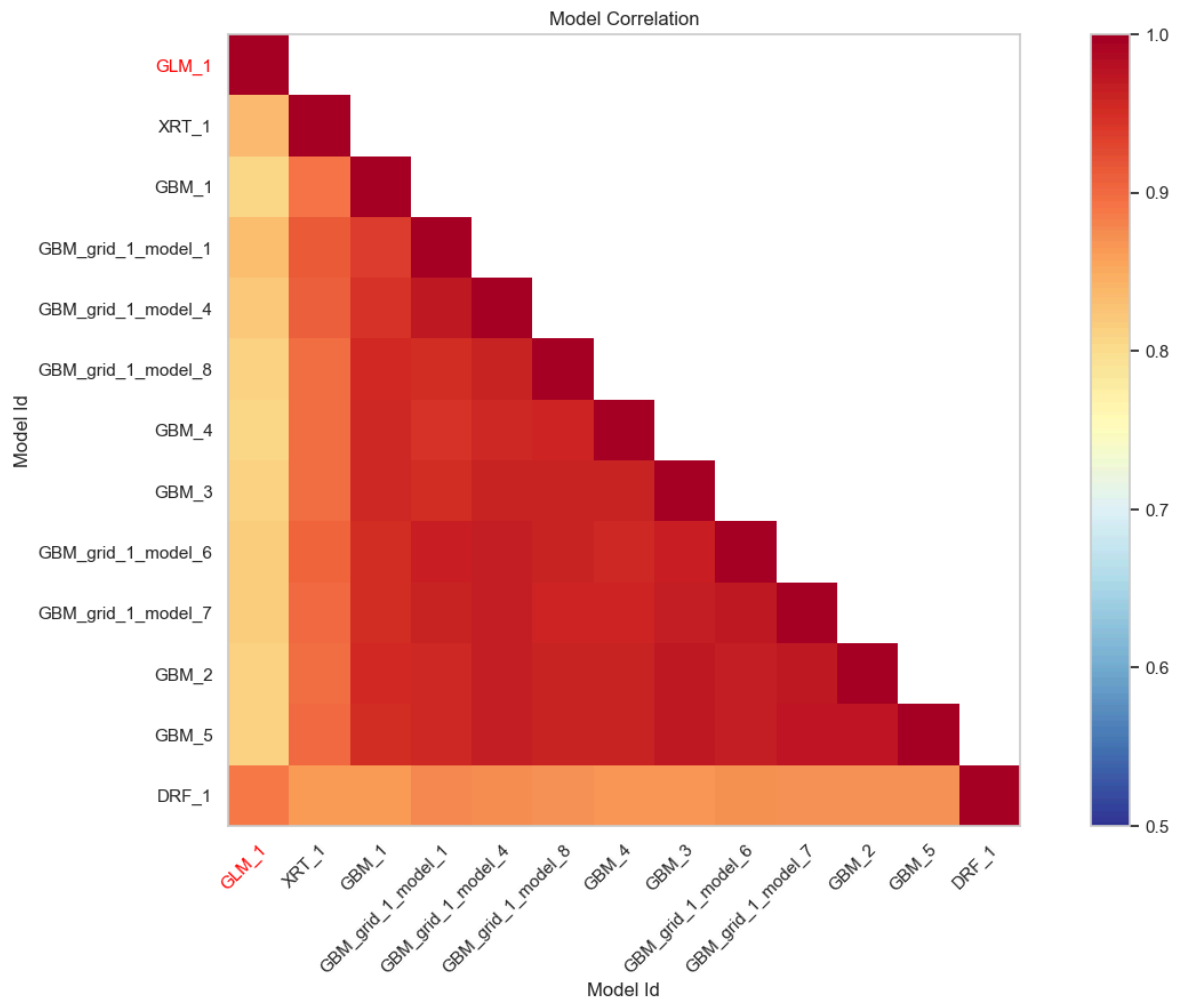
Variable Importance Heatmap

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.



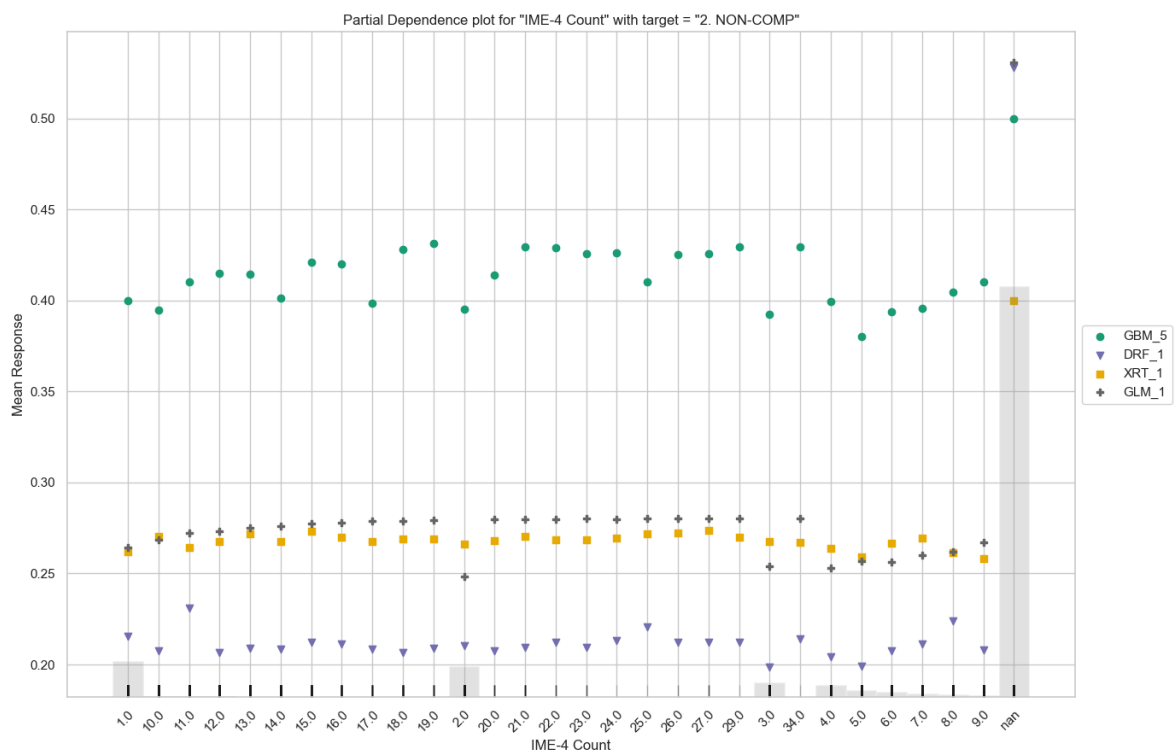
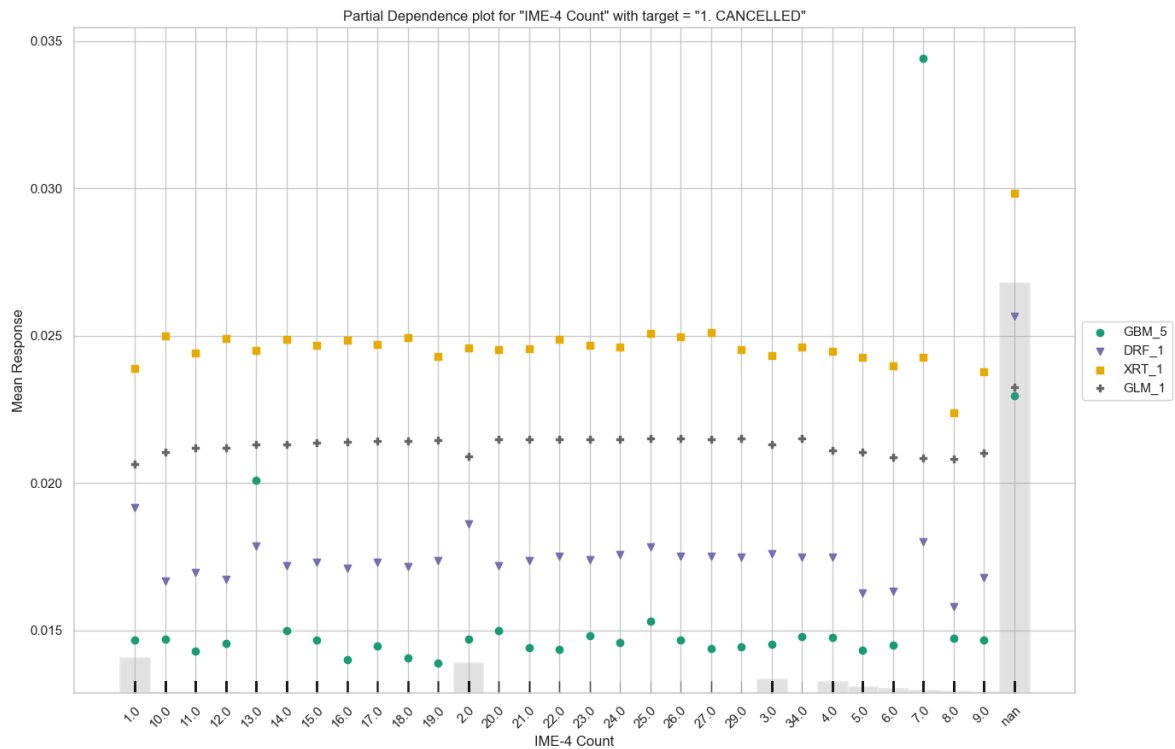
Model Correlation

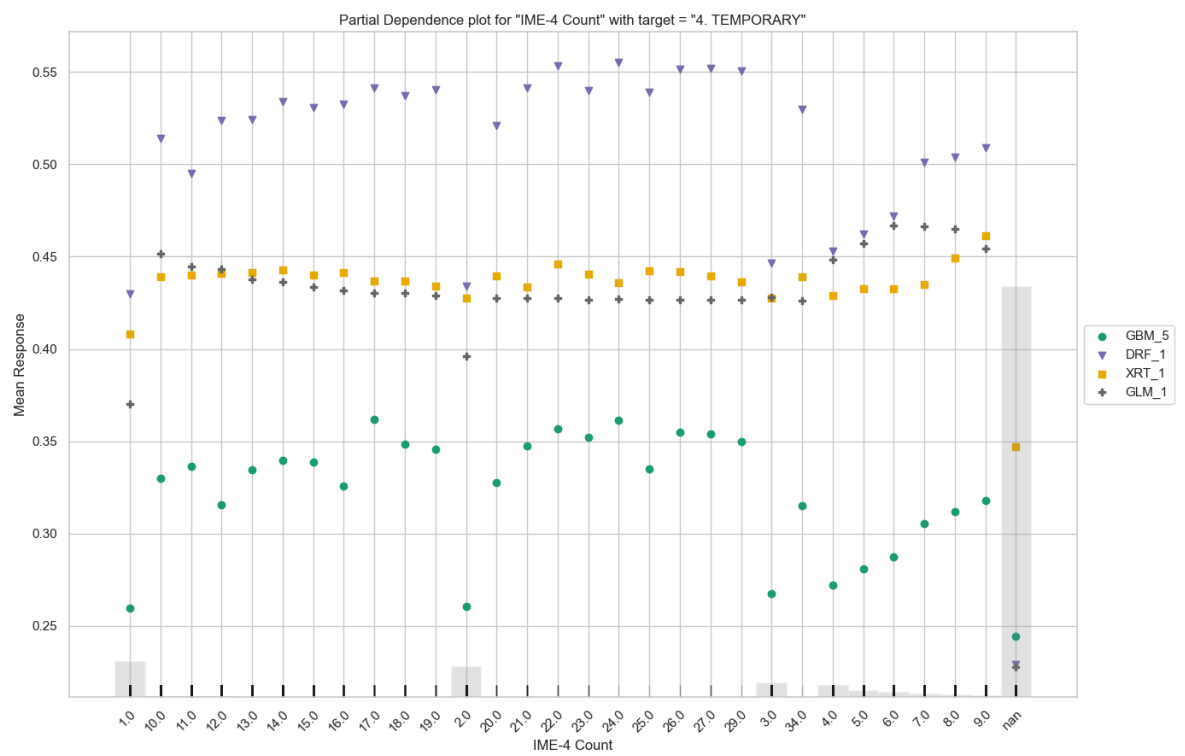
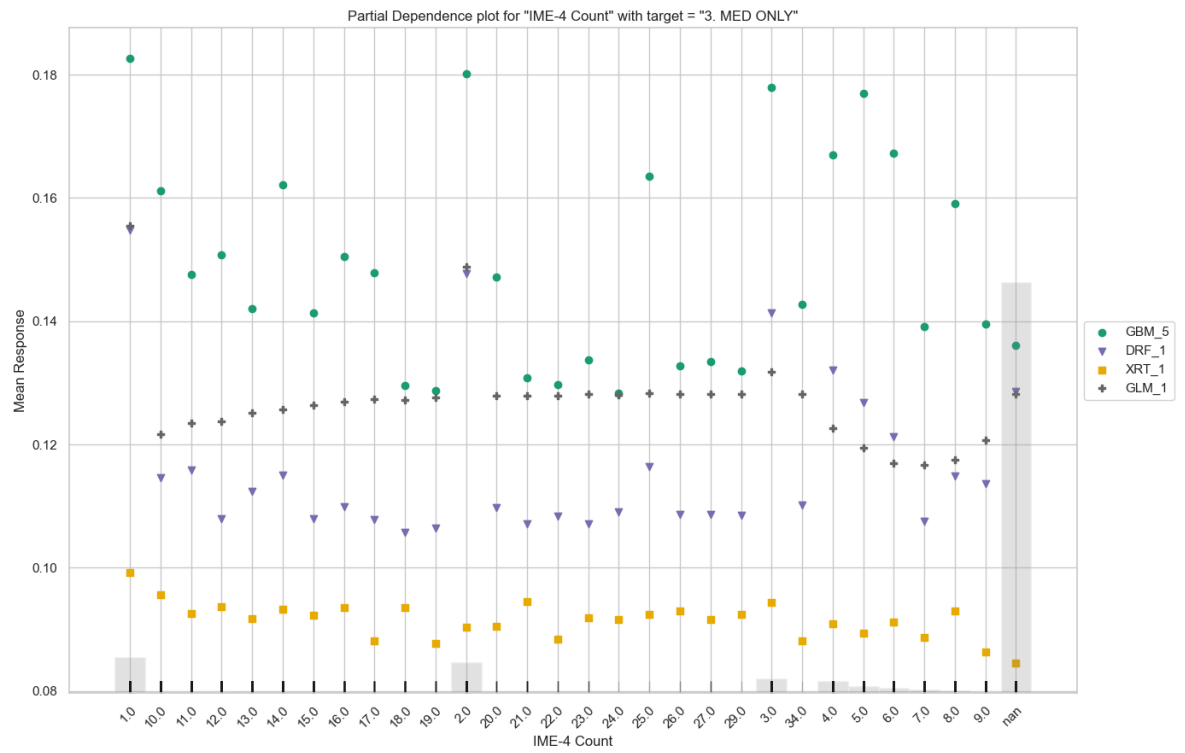
This plot shows the correlation between the predictions of the models. For classification, frequency of identical predictions is used. By default, models are ordered by their similarity (as computed by hierarchical clustering). Interpretable models, such as GAM, GLM, and RuleFit are highlighted using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

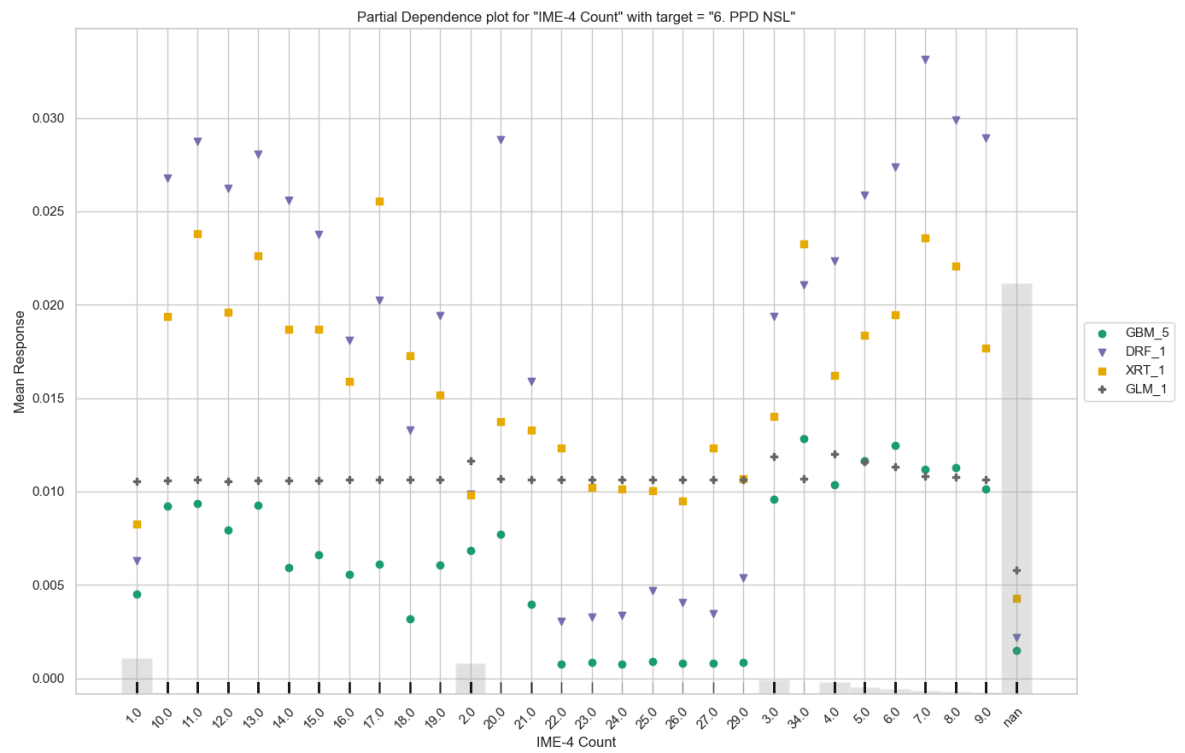
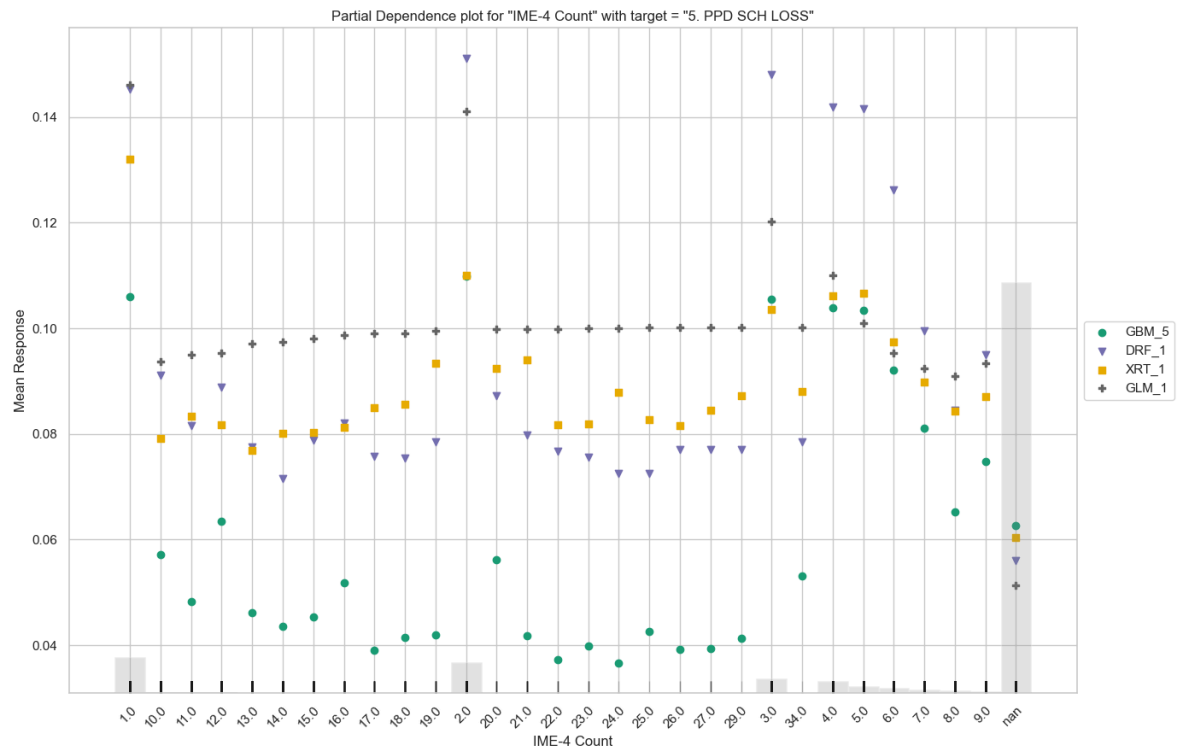


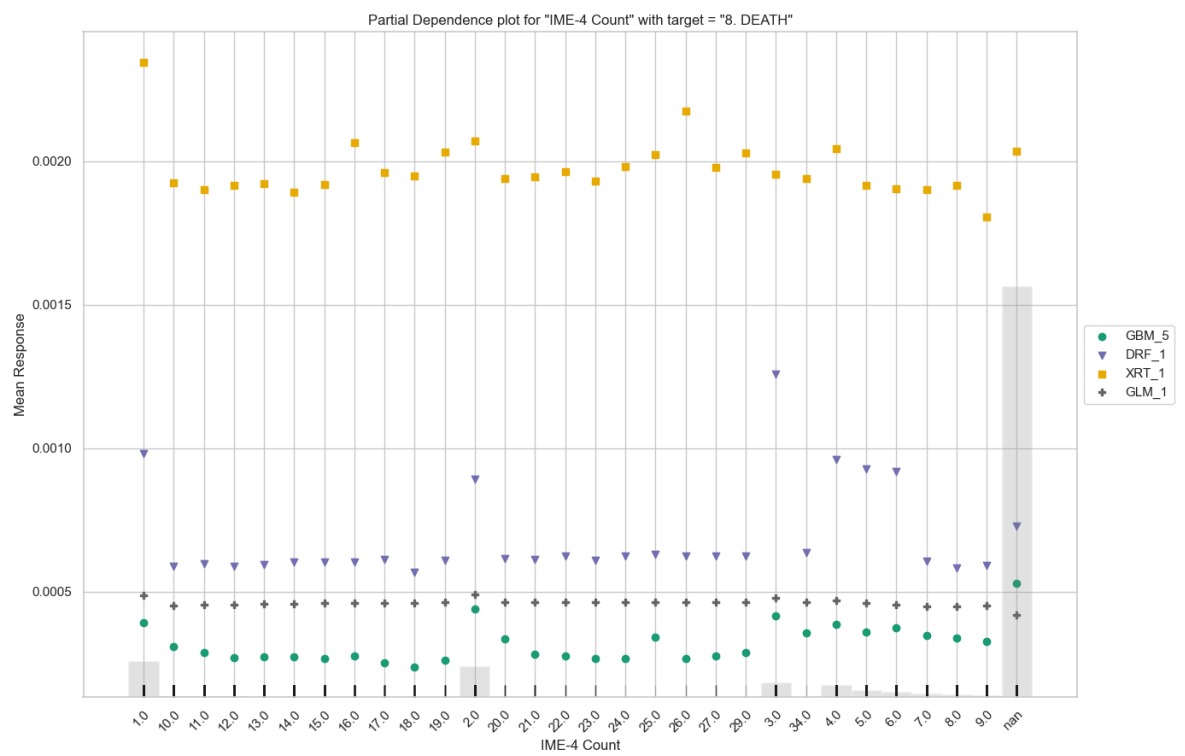
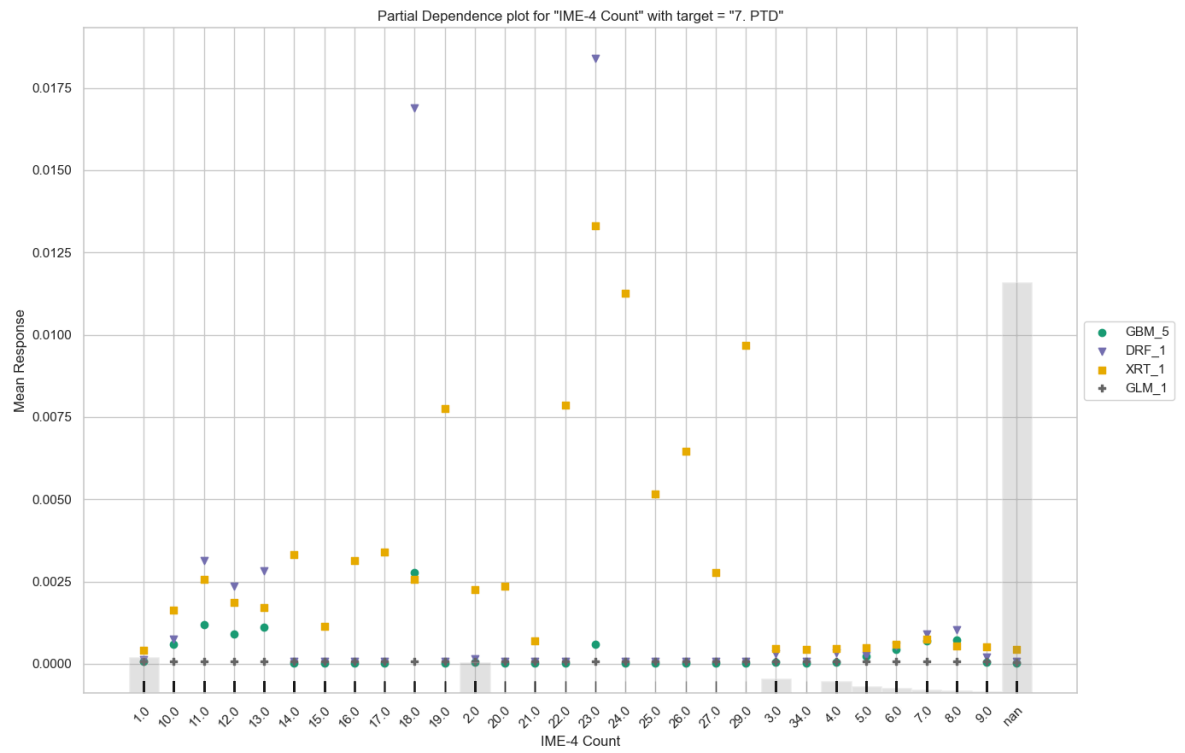
Partial Dependence Plots

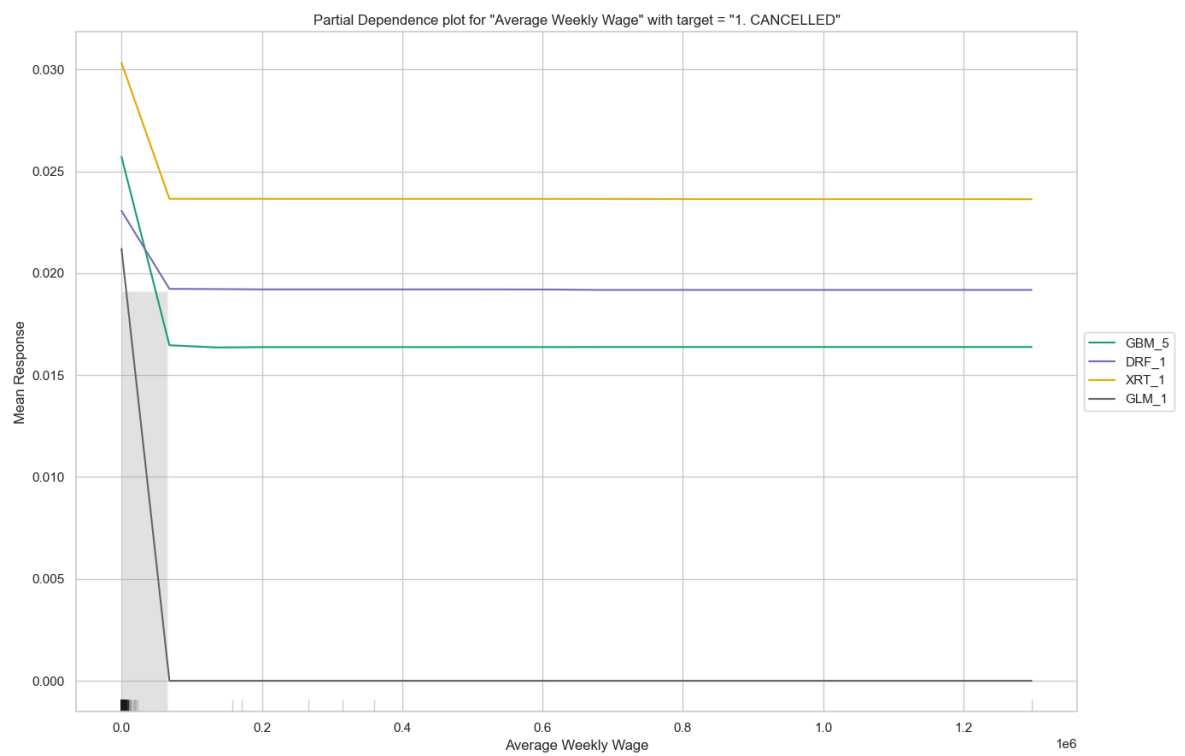
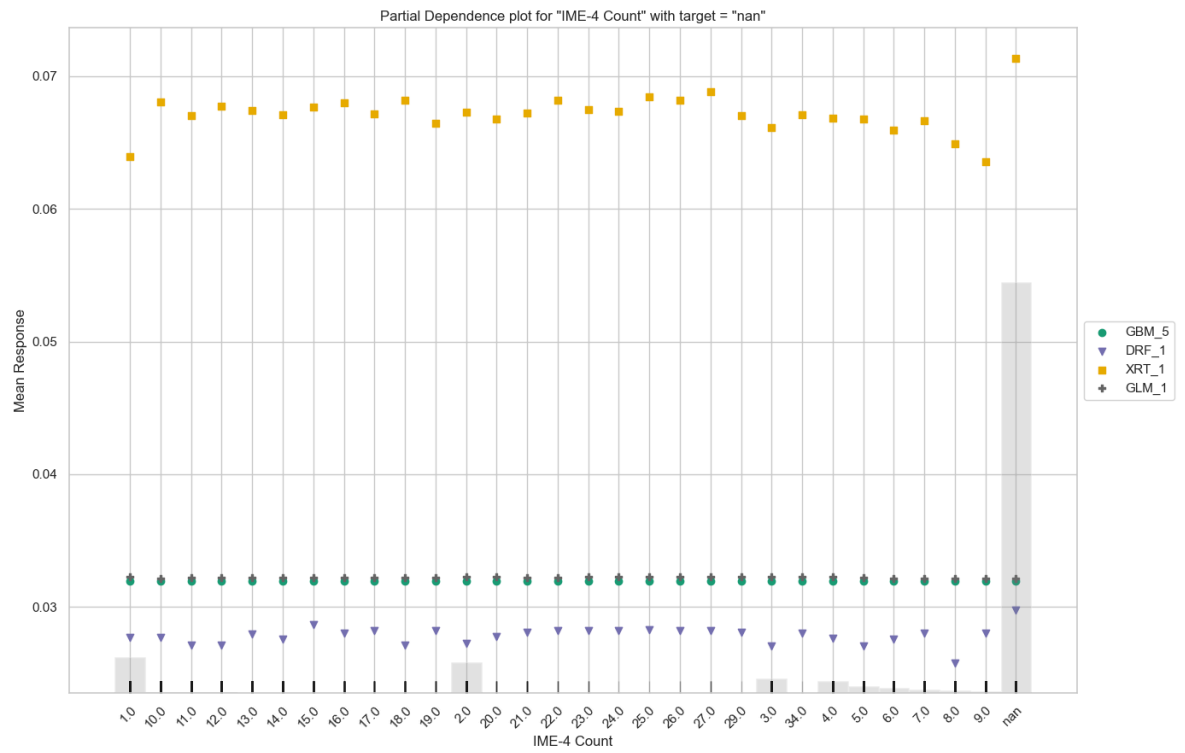
Partial dependence plot (PDP) gives a graphical depiction of the marginal effect of a variable on the response. The effect of a variable is measured in change in the mean response. PDP assumes independence between the feature for which is the PDP computed and the rest.











```

-----
H2OJobCancelled                                     Traceback (most recent call last)
Cell In[31], line 1
----> 1 aml.explain(validation_frame)

File c:\Users\alexg\AppData\Local\Programs\Python\Python38\lib\site-packages\h2o
\explanation\explain.py:3304, in explain(models, frame, columns, top_n_features,
include_explanations, exclude_explanations, plot_overrides, figsize, render, qual
itative_colormap, sequential_colormap, background_frame)
    3302 result["pdp"]["plots"][column] = H2OExplanation()
    3303 for target in targets:
-> 3304     pdp = display(pd_multi_plot(
    3305         models, column=column, target=target,
    3306         **_custom_args(plot_overrides.get("pdp"),
    3307                         frame=frame,
    3308                         figsize=figsize,
    3309                         colormap=qualitative_colormap))
    3310     if target is None:
    3311         result["pdp"]["plots"][column] = pdp

File c:\Users\alexg\AppData\Local\Programs\Python\Python38\lib\site-packages\h2o
\explanation\explain.py:1588, in pd_multi_plot(models, frame, column, best_of_fa
mily, row_index, target, max_levels, figsize, colormap, markers, save_plot_path,
show_rug)
    1585 model_ids = _shorten_model_ids([model.model_id for model in models])
    1586 for i, model in enumerate(models):
    1587     tmp = NumpyFrame(
-> 1588         model.partial_plot(frame, cols=[column], plot=False,
    1589                             row_index=row_index, targets=target,
    1590                             nbins=20 if not is_factor else 1 + frame[column
n].nlevels()[0])[0])
    1591     encoded_col = tmp.columns[0]
    1592     if frame.type(column) == "time":

File c:\Users\alexg\AppData\Local\Programs\Python\Python38\lib\site-packages\h2o
\utils\metaclass.py:102, in deprecated_params.<locals>.decorator.<locals>.wrapper
(*args, **kwargs)
    100 for msg in messages:
    101     warnings.warn(msg, H2ODeprecationWarning, 2)
--> 102 return fn(*args, **new_kwargs)

File c:\Users\alexg\AppData\Local\Programs\Python\Python38\lib\site-packages\h2o
\model\model_base.py:1363, in ModelBase.partial_plot(self, frame, cols, destinati
on_key, nbins, weight_column, plot, plot_stddev, figsize, server, include_na, use
r_splits, col_pairs_2dpdp, save_plot_path, row_index, targets)
    1360 kwargs["targets"] = targets
    1362 self._generate_user_splits(user_splits, frame, kwargs)
-> 1363 json = H2OJob(h2o.api("POST /3/PartialDependence/", data=kwargs), job_ty
pe="PartialDependencePlot").poll()
    1364 json = h2o.api("GET /3/PartialDependence/%s" % json.dest_key)
    1366 # Extract partial dependence data from json response

File c:\Users\alexg\AppData\Local\Programs\Python\Python38\lib\site-packages\h2o
\job.py:85, in H2OJob.poll(self, poll_updates)
    83 # check if failed... and politely print relevant message
    84 if self.status == "CANCELLED":
---> 85     raise H2OJobCancelled("Job<%s> was cancelled by the user." % self.job
_key)
    86 if self.status == "FAILED":
    87     if (isinstance(self.job, dict)) and ("stacktrace" in list(self.job)):

```

H2OJobCancelled: Job<\$03017f00000132d4fffffffff\$_af7d41cdeaca6e75156ae2d4bd5a404f>
was cancelled by the user.
<Figure size 1600x900 with 0 Axes>