## Types of models to train

Your final submission should include single model. The model set you should try to come up with best model per type of model:

1. Identify best model from: H2O-3 GLM - try different combinations of regularization

**Evaluation metric: AUCPR**

## Feature engineering

You should train/fit categorical features scalers and encoders on Train only. Use `transform` or equivalent function on Validation/Test datasets.

It is important to understand all the steps before model training, so that you can reliably replicate and test them to produce scoring function.

You should generate various new features. Examples of such features can be seen in the Module-3 lecture on GLMs.
Your final model should have at least **10** new engineered features.
On-hot-encoding, label encoding, and target encoding **is not included in the 10** features to create.
You can attempt target encoding, however the technique is not expected to produce improvement for Linear models.

Ideas for Feature engineering for various types of variables:

1. https://docs.h2o.ai/driverless-ai/1-10-lts/docs/userguide/transformations.html
2. GLM lecture and hands-on (Module-3)

**Note**:

- You don't have to perform feature engineering using H2O-3 even if you decided to use H2O-3 GLM for model training.
- It is OK to perform feature engineering using any technique, as long as you can replicate it correctly in the Scoring function.

## Threshold calculation

You will need to calculate optimal threshold for class assignment using F1 metric:

- Using H2O-3, use F1

You will need to find optimal probability threshold for class assignment, the threshold that maximizes above F1.

# H2O Model

```
import h2o
try:
    h2o.cluster().shutdown()
```

```
except:
    pass
```

```
H2O session _sid_a34c closed.
```

```
from h2o.frame import H2OFrame
h2o.init(max_mem_size = "4G", nthreads=16)
```

```
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 25.361-b09, mixed mode)
  Starting server from D:\Work\Gre\UTD\Courses\Fall\MIS6341\Softwares\
Python\ml-fall-2023\Lib\site-packages\h2o\backend\bin\h2o.jar
  Ice root: C:\Users\Asus\AppData\Local\Temp\tmpb9pjpbvq
  JVM stdout: C:\Users\Asus\AppData\Local\Temp\tmpb9pjpbvq\
h2o_Asus_started_from_python.out
  JVM stderr: C:\Users\Asus\AppData\Local\Temp\tmpb9pjpbvq\
h2o_Asus_started_from_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:         08 secs
H2O_cluster_timezone:       America/Chicago
H2O_data_parsing_timezone:  UTC
H2O_cluster_version:        3.42.0.3
H2O_cluster_version_age:    2 months and 14 days
H2O_cluster_name:           H2O_from_python_Asus_zbkukp
H2O_cluster_total_nodes:    1
H2O_cluster_free_memory:    3.556 Gb
H2O_cluster_total_cores:    16
H2O_cluster_allowed_cores:  16
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:             3.10.11 final
--------------------------  -----------------------------
```

We import an H2O DataFrame (df_h) from a CSV file located at the specified path and then display its first few rows using df_h.head(). It is imported like a a pandas data frame, only difference being that it resides inside a cluster

```
from h2o.estimators.glm import H2OGeneralizedLinearEstimator

# Specify the path to your CSV file
csv_file_path =
'D:/Work/Gre/UTD/Courses/Fall/MIS6341/Softwares/Python/ml-fall-2023/
Project1/h2o_df.csv'
```

```
df_h = h2o.import_file(csv_file_path)

df_h.head()

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

     City      State     Zip       Bank     BankState     NAICS
NoEmp    NewExist    CreateJob    RetainedJob    FranchiseCode
UrbanRural    RevLineCr     LowDoc    DisbursementGross
BalanceGross      GrAppv     SBA_Appv    MIS_Status
Log_DisbursementGross    Log_GrAppv    Log_SBA_Appv
Log_BalanceGross    TotalJobs    IncomeToLoanRatio
EmployeesToLoanRatio    JobPerLoan    Gauren_SBA_Appv    DefaultRate
---------  ---------  -----  ---------  -----------  -------
----------  ----------  -----------  ------------  ---------------
------------  -----------  ---------  ------------------
-------------  ----------  -----------  ------------
----------------------  ------------  -------------
-----------------  ----------  ------------------
----------------------  -----------  -----------------  ------------
0.11465    0.184773   93001  0.0314465    0.218517    235910
0.600407     0.17044    -0.0353733     -0.0454543               1
0.0716743     0.15307   0.187063             0.358949      -0.00229552
0.394801    0.410973              0              0.306712
0.332752       0.344279           -0.00229816  -0.0808276
0.873414              1.46094     -0.196674            0.960651
17.5096
0.137597   0.165992   44039  0.128698     0.159167    484121  -0.1536
0.17044    -0.0353733     -0.0454543               0     0.243491
0.15307   0.187063            -0.614207      -0.00229552  -0.594552
-0.600302              0             -0.952455      -0.902762      -
0.917047           -0.00229816  -0.0808276             1.02316
0.255872     0.134645             0.990421       17.5096
0.139151   0.116799   68122  0.175694     0.159167    451120
0.115688     0.17044    -0.0139777      0.0185835           38510
0.243491      0.15307   0.187063             2.31704       -0.00229552
2.752       2.53323               0              1.19907
1.32229        1.26221            -0.00229816   0.00460579
0.914656              0.045668     0.00181815          1.08636
17.5096
0.140704   0.197662   14208  0.118949     0.167297    321114   0.51962
0.17044    -0.0353733     -0.0454543               1     0.0716743
0.15307   0.187063            -0.0046858     -0.00229552   0.0251141
0.00123195            0             -0.00469681     0.0248039
0.00123119         -0.00229816  -0.0808276            -3.80356
421.787      -65.6095              20.3856          17.5096
0.140354   0.144273   16335  0.193277     0.0783071        0
```

```
0.102223     0.17044    -0.0353733     -0.0454543                1
0.0716743     0.15307   0.187063             0.584056      -0.00229552
0.623655    0.637638              0               0.459989
0.48468        0.493255           -0.00229816  -0.0808276
0.915969              0.160316    -0.126761             0.978071
17.5096
0.164738   0.188249   77381  0.141056     0.180023    532490  -
0.140136     0.17044    -0.0268151     -0.0411851                1
0.243491      0.251569  0.187063           -0.607811      -0.00229552
-0.608635   -0.60902               0              -0.93601       -
0.938115      -0.939099           -0.00229816  -0.0680002
0.998014              0.230101     0.111655             0.999368
17.5096
0.172811   0.223814   48334  0.214858     0.198182    811111  -
0.0458851    0.17044    -0.0353733     -0.0113008                0
0.243491      0.251569  0.187063           -0.584843      -0.00229552
-0.65335    -0.636699              0              -0.879098      -
1.05944       -1.01252            -0.00229816  -0.0466741
0.918554              0.0720672    0.0733063            1.02615
17.5096
0.275041   0.184773   90016  0.182257     0.180023    453210  -
0.0997428    0.17044    -0.0353733     -0.0454543            50564
0.0716743     0.15307   0.0897581          -0.437584      -0.00229552
-0.414989   -0.391073              1              -0.575514      -
0.536125      -0.496057           -0.00229816  -0.0808276
1.11893               0.255049     0.206682             1.06116
17.5096
0.170819   0.179137   19805  0.214858     0.198182    722110  -
0.113207     0.17044    -0.0353733     -0.0326467                1
0.243491      0.251569  0.187063           -0.680008      -0.00229552
-0.661447   -0.641712              0              -1.13946       -
1.08308       -1.02642            -0.00229816  -0.06802
1.05968               0.176414     0.105998             1.03075
17.5096
0.0820406  0.0793893  85267  0.329864     0.0762233   484220  -
0.140136     0.186978   -0.0310942     -0.0369159                1
0.187265      0.15307   0.187063           -0.664423      -0.00229552
-0.645604   -0.631905              0              -1.0919        -
1.03734       -0.999413           -0.00229816  -0.0680101
1.05146               0.221768     0.107627             1.02168
17.5096
[10 rows x 29 columns]
```

```
df_h.describe()
```

```
Rows:566472
Cols:29

         City                  State                Zip
Bank                     BankState           NAICS               NoEmp
```

```
NewExist              CreateJob              RetainedJob
FranchiseCode       UrbanRural           RevLineCr             LowDoc
DisbursementGross       BalanceGross          GrAppv
SBA_Appv               MIS_Status           Log_DisbursementGross
Log_GrAppv            Log_SBA_Appv          Log_BalanceGross
TotalJobs               IncomeToLoanRatio    EmployeesToLoanRatio
JobPerLoan             Gauren_SBA_Appv     DefaultRate
-------  --------------------  -------------------  -----------------
--------------------  -------------------  ------------------
---------------------  --------------------  ---------------------
-----------------------  ------------------  -------------------
--------------------  --------------------  ----------------------
---------------------  ----------------------  ---------------------
-------------------  -----------------------  --------------------
--------------------  ----------------------  ----------------------
-------------------  ----------------------  ---------------------
------------------  ---------------------
type     real                  real                  int
real                  real                 int                  real
real                  real                   real
int                 real                 real                  real
real                    real                   real
real                   int                  real
real                  real                  real
real                    real                   real
real                   real                 real
mins     0.002343996205589325  0.07006369426751592  0.0
0.0                   0.05249080575058509  0.0                  -
0.153600434175742     0.17044022221184818   -0.0353732780628916 5   -
0.04545428677284537     0.0                  0.07167428097535124
0.1530699251248866    0.08975812818488481   -0.6973236731572443     -
0.002295517679303139  -0.6783474528695129     -0.6521736527433548
0.0                  -1.195091272676935       -1.1342833622057995   -
1.056051925886739    -0.002298156418966926   -0.08082756483573703    -
3249.7995325776096  -6217.269704293639      -6946.62619277787      -
1425.410980018344  17.509603299015662
mean     0.17567877828071593   0.1750960329901566   53853.91636479826
0.17619418541215054   0.17509732398068 8    398590.7101374825
8.12805739334945e-18   0.1750960329901566    7.224939905199511e-18  -
5.519051316471848 6e-18  2748.614023288001   0.17509603299015666
0.17509603299015658   0.17509603299015658   -1.204156650866585e-18
5.218012153755202e-18  1.7058885887276622e-17  -6.70313868982399e-17
0.1750960329901566   -0.2785689460749845      -0.279916152561005    -
0.27948106942859474  -0.002248711799015892 5  1.0636717082654834e-17
2.112198455611862    0.4918000867212119      -0.2489405491913917
1.496274454678428   17.509603299015662
maxs     0.6626422957411695    0.27204321821338995  99999.0
0.8892988929799549    0.3800327332242226   928120.0
134.4770591134275      0.1869783599731616    37.62078512466269
```

```
40.51182843681707        92006.0             0.2434914068009948
0.25156900049734354   0.18706258164639383   37.39775945774547
552.889522851626       18.58692184996103       23.199528251630852
1.0                   3.6479991104368574       2.9748620909714023
3.186333139238373     6.316965249625691       75.14418220289946
6396.558729319238    37035.12034202456       11627.399763221902
1043.390760820282    17.509603299015662
sigma    0.06639769763247672   0.04199181924329759  31185.22931842036
0.12227701398294173   0.07450052965800262  263321.75739069114
1.000000882657328      0.007437873600660589  1.0000008826573281
1.0000008826573283       12747.306218543337  0.07945017170394587
0.041041501003144853  0.031956251232178115  1.0000008826573277
1.00000088265733       1.0000008826573281      1.0000008826573283
0.38004955887207637  0.6751342455877893       0.673495391264487
0.6724602419470982    0.015557287489121684    1.9967248008580234
48.40854150735596    57.47035513066714       29.416621353104784
11.623336619659533  4.396234168225107e-15
zeros    0                     0                    176
20864                 0                    127266              0
0                     0                      0
131369              0                    0                     0
0                       0                      0
0                      467285               0                     0
0                     0                       0
0                    0                       0                     0
0
missing  0                     0                    0
0                     0                    0                   0
0                     0                      0
0                   0                    0                     0
0                       0                      0
0                      0                    0                     0
0                     0                       0
0                    0                       0                     0
0
0        0.11464974937637376   0.18477258385515263  93001.0
0.031446540880519576  0.21851689918783204  235910.0
0.6004066600002538     0.17044022221184818   -0.03537327806289165   -
0.04545428677284537     1.0                  0.07167428097535124
0.1530699251248866    0.18706258164639383   0.35894908638142187     -
0.002295517679303139  0.394801069280922       0.4109726208502541
0.0                  0.3067116704358818       0.3327518024385963
0.3442792686074715    -0.002298156418966926   -0.08082756483573703
0.873413624583551    1.4609407769259248      -0.19667384330497317
0.9606505378974516  17.509603299015662
1        0.137596588115639     0.1659919028340081   44039.0
0.12869797760320908   0.15916691590309687  484121.0            -
0.153600434175742     0.17044022221184818   -0.03537327806289165   -
0.04545428677284537     0.0                  0.2434914068009948
```

```
0.1530699251248866    0.18706258164639383   -0.6142071281443656     -
0.002295517679303139  -0.5945517349325695     -0.6003022273404686
0.0                  -0.9524546549496404      -0.9027619966303708   -
0.9170465858109552   -0.002298156418966926   -0.08082756483573703
1.0231631671025114   0.25587183785114576     0.1346447858336776
0.99042067121194    17.509603299015662
2        0.1391509433962264    0.11679920477137178  68122.0
0.1756944756092088    0.15916691590309687  451120.0
0.11568781374425648    0.17044022221184818   -0.013977733515888482
0.01858352805387556      38510.0              0.2434914068009948
0.1530699251248866    0.18706258164639383   2.3170363593098213      -
0.002295517679303139  2.751995739818049       2.533233145474642
0.0                  1.199071721409156        1.3222878957003827
1.2622133569935532    -0.002298156418966926   0.00460579453798708
0.9146557881768466   0.045668048340090896    0.0018181486951624856
1.0863570708974033  17.509603299015662
3        0.1407035175879397    0.19766159276417383  14208.0
0.1189488243430152    0.16729655837889998  321114.0
0.5196201856242543     0.17044022221184818   -0.03537327806289165   -
0.04545428677284537     1.0                 0.07167428097535124
0.1530699251248866    0.18706258164639383   -0.004685798049922222   -
0.002295517679303139  0.02511407838264226     0.0012319496005647338
0.0                  -0.0046968108174749315   0.024803902380498698
0.00123119137332351   -0.002298156418966926   -0.08082756483573703
-3.803563106619152   421.7868859132362       -65.60947363324371
20.385637830581544  17.509603299015662
4        0.14035375121577756   0.14427252985884909  16335.0
0.19327731092436976   0.07830707560361704  0.0
0.10222340134825655    0.17044022221184818   -0.03537327806289165   -
0.04545428677284537     1.0                 0.07167428097535124
0.1530699251248866    0.18706258164639383   0.5840563957913015      -
0.002295517679303139  0.6236549207893809      0.6376376730309333
0.0                  0.4599888961620474       0.4846797319622258
0.4932547601265484    -0.002298156418966926   -0.08082756483573703
0.91596908478613     0.16031581205412476     -0.12676096199199305
0.9780710067284966  17.509603299015662
5        0.16473815924050095   0.1882487805974511   77381.0
0.1410558507971523    0.18002315611603092  532490.0            -
0.14013602177974208   0.17044022221184818   -0.02681506024409038   -
0.041185099117730634    1.0                 0.2434914068009948
0.25156900049734354   0.18706258164639383   -0.6078106173677496     -
0.002295517679303139  -0.6086350488715516     -0.6090201139628024
0.0                  -0.936010436896243       -0.9381147754788066   -
0.9390991626800832   -0.002298156418966926   -0.06800015936182102
0.998014028490483    0.23010081041149533     0.11165503043791804
0.9993677300922869  17.509603299015662
6        0.1728110599078341    0.22381434467720585  48334.0
0.21485771162285663   0.1981815002622836   811111.0            -
0.045885135007742606  0.17044022221184818   -0.03537327806289165   -
```

```
0.011300785531927536    0.0                 0.2434914068009948
0.25156900049734354   0.18706258164639383   -0.5848427454291907     -
0.002295517679303139  -0.6533495706278197     -0.6366994039887123
0.0                  -0.8790979038493719     -1.059438414961258     -
1.012524699207778    -0.002298156418966926   -0.04667406359481919
0.9185539389001204   0.07206718699638688     0.07330627813128383
1.026150749529212   17.509603299015662
7        0.27504127682993945   0.18477258385515263  90016.0
0.182256711409396     0.18002315611603092  453210.0           -
0.0997427845917423    0.17044022221184818   -0.03537327806289165    -
0.04545428677284537     50564.0             0.07167428097535124
0.1530699251248866    0.08975812818488481   -0.4375844699919985     -
0.002295517679303139  -0.4149894822105479     -0.391072948404457
1.0                  -0.5755143250547002     -0.5361254527846345    -
0.4960568023622707   -0.002298156418966926   -0.08082756483573703
1.1189331089693224   0.25504905158662605     0.20668155433789612
1.0611561958035405  17.509603299015662
8        0.1708185053380783    0.17913669064748203  19805.0
0.21485771162285663   0.1981815002622836   722110.0           -
0.11320719698774225   0.17044022221184818   -0.03537327806289165    -
0.03264672380750118     1.0                 0.2434914068009948
0.25156900049734354   0.18706258164639383   -0.6800077262795612     -
0.002295517679303139  -0.6614474761427344     -0.6417121887965542
0.0                  -1.1394584281034796     -1.0830760321756987    -
1.0264186736605694   -0.002298156418966926   -0.06802000187039284
1.0596771234076527   0.17641428503960202     0.10599767786545444
1.0307541101614277  17.509603299015662
9        0.0820405814622732    0.07938931297709924  85267.0
0.32986399789891396   0.07622333751568382  484220.0           -
0.14013602177974208   0.1869783599731616    -0.031094169153491016  -
0.03691591146261591     1.0                 0.18726450640542577
0.1530699251248866    0.18706258164639383   -0.6644233740896465     -
0.002295517679303139  -0.6456037479613795     -0.6319045663464287
0.0                  -1.091904955472731      -1.0373396358189255    -
0.9994130438532118   -0.002298156418966926   -0.06801008061610692
1.0514615805535896   0.2217676991796185      0.10762713903039244
1.0216791938918204  17.509603299015662
[566472 rows x 29 columns]
```

To check if the data frame is indeed H2OFrame

```
if isinstance(df_h, h2o.H2OFrame):
    print('It is H2O data frame')
else:
    print('It is not H2O data frame')

It is H2O data frame
```

The provided code splits an H2O DataFrame into training, validation, and test sets and separates predictor columns from the response column for machine learning tasks.

The first number, 0.7, specifies that 70% of the data will be used for training (the "train" subset).

The second number, 0.15, specifies that 15% of the data will be used for validation (the "valid" subset).

The remaining 15% not specified is implicitly used for testing (the "test" subset).

The seed=1234 parameter is used to set the random seed for reproducibility, ensuring that the same split is obtained when the code is executed multiple times.

```
# Split the data as described above
train, valid, test = df_h.split_frame([0.7, 0.15], seed=1234)

# Prepare predictors and response columns
train_X = df_h.columns
train_y = "MIS_Status"
train_X.remove(train_y)
```

Since we already imported the H2O GLM estimator, we will just instantiate our model. For simplicity, the name of our model will be glm. To build a GLM, you just need to define the family, and you are ready to go. However, we will set a random seed for reproducibility purposes, and also a model id to be able to retrieve the model later on if we need to access it. You can instantiate your GLM, as shown below.

```
glm = H2OGeneralizedLinearEstimator(family = "binomial", seed = 42,
model_id = 'default_glm')

%time glm.train(x = train_X, y = train_y, training_frame = train,
validation_frame = valid)

glm Model Build progress: |

████████████████████████████████████████████████████████| (done) 100%
CPU times: total: 15.6 ms
Wall time: 1.73 s

Model Details
=============
H2OGeneralizedLinearEstimator : Generalized Linear Modeling
Model Key: default_glm


GLM Model: summary
    family    link    regularization
number_of_predictors_total    number_of_active_predictors
number_of_iterations    training_frame
--  --------  ------  -----------------------------------------
----------------------------  ------------------------------
```

```
---------------------  ----------------
    binomial  logit   Elastic Net (alpha = 0.5, lambda = 2.505E-4 )
27                            22                             4
py_17_sid_b6e5

ModelMetricsBinomialGLM: glm
** Reported on train data. **

MSE: 0.12059947193887977
RMSE: 0.34727434679066027
LogLoss: 0.38760076479874667
AUC: 0.7768657635596997
AUCPR: 0.4455013625622545
Gini: 0.5537315271193994
Null degrees of freedom: 396643
Residual degrees of freedom: 396621
Null deviance: 367114.9304608208
Residual deviance: 307479.03550566814
AIC: 307525.03550566814

Confusion Matrix (Act/Pred) for max f1 @ threshold =
0.22939337808642146
       0       1      Error    Rate
-----  ------  -----  -------  ------------------
0      267868  59606  0.182    (59606.0/327474.0)
1      29773   39397  0.4304   (29773.0/69170.0)
Total  297641  99003  0.2253   (89379.0/396644.0)

Maximum Metrics: Maximum metrics at their respective thresholds
metric                       threshold    value     idx
---------------------------  -----------  --------  -----
max f1                       0.229393     0.468529  225
max f2                       0.121245     0.596597  301
max f0point5                 0.345963     0.462715  165
max accuracy                 0.509815     0.83613   100
max precision                0.940776     0.943396  2
max recall                   0.00176269   1         399
max specificity              0.972444     0.999997  0
max absolute_mcc             0.241421     0.341047  218
max min_per_class_accuracy   0.171115     0.703744  264
max mean_per_class_accuracy  0.168308     0.706072  266
max tns                      0.972444     327473    0
max fns                      0.972444     69159     0
max fps                      0.00176269   327474    399
max tps                      0.00176269   69170     399
max tnr                      0.972444     0.999997  0
max fnr                      0.972444     0.999841  0
max fpr                      0.00176269   1         399
max tpr                      0.00176269   1         399
```

```
Gains/Lift Table: Avg response rate: 17.44 %, avg score: 17.44 %
group    cumulative_data_fraction    lower_threshold    lift
cumulative_lift    response_rate    score
cumulative_response_rate    cumulative_score    capture_rate
cumulative_capture_rate    gain      cumulative_gain
kolmogorov_smirnov
-------  -------------------------  ----------------  --------
----------------  --------------  ---------
--------------------------  -----------------  --------------
-------------------------  --------  ----------------
--------------------
1        0.0100014                   0.734962           4.23823
4.23823            0.739098         0.801263   0.739098
0.801263            0.0423883       0.0423883                  323.823
323.823            0.0392278
2        0.0200003                   0.654405           3.70722
3.97276            0.646495         0.693539   0.692802
0.747407            0.0370681       0.0794564                  270.722
297.276            0.0720146
3        0.0300017                   0.596196           3.30877
3.75141            0.57701          0.62454    0.654202
0.706448            0.0330924       0.112549                   230.877
275.141            0.0999829
4        0.0400006                   0.547301           3.15779
3.60302            0.550681         0.570568   0.628325
0.672482            0.0315744       0.144123                   215.779
260.302            0.126116
5        0.050002                    0.506656           2.94017
3.47044            0.51273          0.52673    0.605203
0.643329            0.0294058       0.173529                   194.017
247.044            0.149619
6        0.100002                    0.387661           2.54795
3.00921            0.444332         0.440826   0.52477
0.54208             0.127396        0.300925                   154.795
200.921            0.243363
7        0.150001                    0.314515           2.14662
2.72168            0.374344         0.348613   0.474629
0.477592            0.10733         0.408255                   114.662
172.168            0.312803
8        0.200001                    0.264456           1.76581
2.48272            0.307937         0.28817    0.432956
0.430237            0.0882897       0.496545                   76.5812
148.272            0.359181
9        0.299999                    0.199768           1.3652
2.11022            0.238075         0.229392   0.367996
0.363289            0.136519        0.633063                   36.5201
111.022            0.403415
10       0.400001                    0.156437           1.06143
1.84801            0.1851           0.177086   0.322272
```

```
0.316738            0.106144        0.739208                          6.14268
84.8015             0.410855
11       0.5                         0.122838           0.79862
1.63814             0.13927         0.13907    0.285672
0.281205            0.0798612       0.819069                          -20.138
63.8138             0.386464
12       0.599999                    0.0957864          0.610386
1.46685             0.106444        0.108834   0.255801
0.252476            0.061038        0.880107                          -
38.9614  46.6847            0.339273
13       0.700001                    0.0740264          0.484596
1.32652             0.0845078       0.0845272  0.23133
0.228483            0.0484603       0.928567                          -
51.5404  32.6524            0.276845
14       0.799999                    0.0569197          0.357239
1.20536             0.0622983       0.0652598  0.210201
0.208081            0.0357236       0.964291                          -
64.2761  20.5364            0.198993
15       0.899998                    0.0410503          0.249099
1.09911             0.0434399       0.0490687  0.191672
0.190413            0.0249096       0.989201                          -
75.0901  9.91135            0.108044
16       1                           4.27023e-11        0.107993  1
0.0188327        0.0301829  0.174388                    0.17439
0.0107995       1                         -89.2007  0
0

ModelMetricsBinomialGLM: glm
** Reported on validation data. **

MSE: 0.12231119069488473
RMSE: 0.3497301684082812
LogLoss: 0.3920168718665924
AUC: 0.7738999913401364
AUCPR: 0.43815403112953155
Gini: 0.5477999826802729
Null degrees of freedom: 84950
Residual degrees of freedom: 84928
Null deviance: 79045.0523443093
Residual deviance: 66604.45056387779
AIC: 66650.45056387779

Confusion Matrix (Act/Pred) for max f1 @ threshold =
0.2239578972008154
       0      1      Error    Rate
-----  -----  -----  -------  -----------------
0      56619  13383  0.1912   (13383.0/70002.0)
1      6390   8559   0.4275   (6390.0/14949.0)
Total  63009  21942  0.2328   (19773.0/84951.0)
```

```
Maximum Metrics: Maximum metrics at their respective thresholds
metric                       threshold    value     idx
---------------------------  -----------  --------  -----
max f1                       0.223958     0.464016  226
max f2                       0.1221       0.59665   299
max f0point5                 0.362662     0.457985  155
max accuracy                 0.536131     0.833998  89
max precision                0.877112     0.807229  8
max recall                   0.00166505   1         399
max specificity              0.964332     0.999986  0
max absolute_mcc             0.223958     0.331791  226
max min_per_class_accuracy   0.169712     0.702187  262
max mean_per_class_accuracy  0.165484     0.703584  265
max tns                      0.964332     70001     0
max fns                      0.964332     14949     0
max fps                      0.00166505   70002     399
max tps                      0.00166505   14949     399
max tnr                      0.964332     0.999986  0
max fnr                      0.964332     1         0
max fpr                      0.00166505   1         399
max tpr                      0.00166505   1         399

Gains/Lift Table: Avg response rate: 17.60 %, avg score: 17.43 %
group    cumulative_data_fraction    lower_threshold    lift
cumulative_lift    response_rate    score
cumulative_response_rate    cumulative_score    capture_rate
cumulative_capture_rate    gain      cumulative_gain
kolmogorov_smirnov
-------  --------------------------  -----------------  --------
-----------------  ---------------  ---------
--------------------------  ------------------  --------------
-------------------------  --------  -----------------
--------------------
1        0.0100058                   0.737042           4.09824
4.09824           0.721176         0.800378   0.721176
0.800378           0.0410061       0.0410061                   309.824
309.824           0.0376205
2        0.0200115                   0.653884           3.64363
3.87094           0.641176         0.693758   0.681176
0.747068           0.0364573       0.0774634                   264.363
287.094           0.0697207
3        0.0300055                   0.597437           3.31994
3.68742           0.584217         0.625696   0.648882
0.706642           0.0331795       0.110643                    231.994
268.742           0.0978575
4        0.0400113                   0.548427           3.01519
3.51931           0.530588         0.571843   0.6193
0.672933           0.0301692       0.140812                    201.519
251.931           0.122327
```

```
5        0.0500053                  0.508851          2.86479
3.3885            0.504122        0.527854   0.596281
0.643937           0.0286307      0.169443                    186.479
238.85            0.144943
6        0.100011                   0.388025          2.53502
2.96176           0.446092        0.44178    0.521186
0.542859           0.126764       0.296207                    153.502
196.176           0.238094
7        0.150004                   0.314092          2.0673
2.66365           0.363786        0.348682   0.468728
0.478143           0.103351       0.399558                    106.73
166.365           0.302847
8        0.200009                   0.265068          1.69893
2.42246           0.298964        0.288529   0.426285
0.430737           0.0849555      0.484514                    69.893
142.246           0.345261
9        0.300008                   0.199691          1.41616
2.08704           0.249205        0.229498   0.36726
0.36366            0.141615       0.626129                    41.6165
108.704           0.395764
10       0.400007                   0.156012          1.07968
1.83521           0.189994        0.176796   0.322945
0.316945           0.107967       0.734096                    7.96836
83.5207           0.405434
11       0.500006                   0.122342          0.814111
1.63099           0.143261        0.13849    0.287009
0.281255           0.0814101      0.815506                    -
18.5889  63.0993          0.382876
12       0.600005                   0.0954449         0.661591
1.46943           0.116421        0.108555   0.258578
0.252472           0.0661583      0.881664                    -
33.8409  46.9429          0.341808
13       0.700004                   0.0737534         0.482981
1.32851           0.0849912       0.0840837  0.233781
0.228417           0.0482975      0.929962                    -
51.7019  32.851           0.279066
14       0.800002                   0.0567782         0.346516
1.20576           0.060977        0.0650276  0.212181
0.207994           0.0346511      0.964613                    -
65.3484  20.5763          0.199763
15       0.900001                   0.0411062         0.253532
1.09996           0.0446145       0.0491226  0.193562
0.190342           0.0253529      0.989966                    -
74.6468  9.99607          0.109177
16       1                          6.97019e-06       0.100342  1
0.0176574       0.0301928  0.175972                   0.174327
0.0100341      1                      -89.9658  0
0
```

```
Scoring History:
    timestamp            duration    iterations
negative_log_likelihood    objective    training_rmse
training_logloss     training_r2         training_auc
training_pr_auc     training_lift      training_classification_error
validation_rmse     validation_logloss    validation_r2
validation_auc      validation_pr_auc    validation_lift
validation_classification_error
--  -------------------  ----------  ------------
-------------------------  ----------  -----------------
-------------------  -------------------  -----------------
------------------  -----------------  ------------------------------
------------------  --------------------  ------------------
------------------  ------------------  ----------------
---------------------------------
    2023-11-05 20:51:48  0.000 sec   0             183557
0.462776
    2023-11-05 20:51:48  0.214 sec   1             156657
0.395782
    2023-11-05 20:51:48  0.285 sec   2             153810
0.388709
    2023-11-05 20:51:48  0.340 sec   3             153734
0.38859
    2023-11-05 20:51:48  0.403 sec   4             153740
0.388591     0.34727434679066027  0.38760076479874667
0.16236930409227046  0.7768657635596997  0.4455013625622545
4.238233490486189  0.22533808654612197            0.3497301684082812
0.3920168718665924    0.15650872440673047  0.7738999913401364
0.43815403112953155  4.098244856039948  0.2327577073842568

Variable Importances:
variable               relative_importance    scaled_importance
percentage
---------------------  ---------------------  ---------------------
---------------------
Log_DisbursementGross  1.0742710828781128     1.0
0.17945711926553892
GrAppv                 0.7878198623657227     0.7333529450081168
0.13160540691605582
Log_GrAppv             0.7507895827293396     0.6988828003429787
0.12541949405378375
DisbursementGross      0.7272646427154541     0.6769842866541813
0.12148964987099521
Log_SBA_Appv           0.5599473714828491     0.5212347054736658
0.0935392787055257
Bank                   0.5510706901550293     0.5129717246773866
0.09205642797527898
City                   0.3853748142719269     0.3587314416389738
0.06437691110650404
```

```
UrbanRural             0.3724907636642456     0.34673814607975306
0.06222462883494611
SBA_Appv               0.1293647140264511     0.12042092176572985
0.021610391719378712
RevLineCr              0.11934459209442139    0.11109355356999986
0.019936529092644
---                    ---                    ---
---
BankState              0.03160914033651352    0.029423802651215834
0.005280310861624921
FranchiseCode          0.025763940066099167   0.023982717655467556
0.004303869422608988
Gauren_SBA_Appv        0.008871283382177353   0.008257956044399915
0.0014819490027494534
Log_BalanceGross       0.004079081583768129   0.0037970691464948828
0.0006814110906820303
JobPerLoan             5.3359181038104e-05    4.967012692471231e-05
8.913657891462553e-06
RetainedJob            0.0                    0.0
0.0
BalanceGross           0.0                    0.0
0.0
TotalJobs              0.0                    0.0
0.0
IncomeToLoanRatio      0.0                    0.0
0.0
EmployeesToLoanRatio   0.0                    0.0
0.0
[27 rows x 4 columns]


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

From the summary results, we can see the GLM performance. We will focus on the Area Under the Curve (AUC), and since we have a very imbalanced dataset, we will be looking at the F1 score. Additionally, we will also take a quick look at the misclassification error and logloss.

From the report, we can look at the metrics on the training and validation data, and we see that the training AUCPR was AUCPR: 0.4455013625622545 while the validation AUCPR: 0.43815403112953155

```AUCPR (Area Under the Precision-Recall Curve) is a metric that measures the trade-off between precision and recall for a binary classification model.

AUCPR summarizes how well a model is at ranking the positive class instances higher than the negative class instances.
```

Precision is a measure of the model's ability to correctly identify positive instances among the instances it predicts as positive. It's the ratio of true positive predictions to the total positive predictions

Recall (or Sensitivity) is a measure of the model's ability to identify all the positive instances correctly. It's the ratio of true positive predictions to the total actual positive instances.

The reported values represent the AUCPR score for your model on both the training and validation datasets:

` ` `

Training AUCPR: 0.4455013625622545> This indicates that on the training dataset, the model achieved an AUCPR of approximately 0.4455. This score reflects how well the model performs on the data it was trained on. It suggests that the model is reasonably effective at ranking and identifying positive instances relative to negative instances within the training dataset.

<Validation AUCPR: 0.43815403112953155> This indicates that on the validation dataset, the model achieved an AUCPR of approximately 0.4382. The validation dataset is a separate dataset that the model did not see during training. The AUCPR score on the validation dataset measures the model's generalization performance. While slightly lower than the training AUCPR, this score still suggests that the model maintains its ability to rank positive instances effectively when applied to new, unseen data.

The goal is to have a model that generalizes well, which means it performs consistently on both the training and validation datasets. In this case, the AUCPR scores are reasonably close, indicating that the model is performing consistently, and it is not exhibiting significant overfitting (a situation where the model fits the training data too closely but performs poorly on new data).

From the report, we can also see the max F1 score as well as all the metrics for our model with their respective thresholds. For the default GLM, we obtained a training F1 score of 0.2293934 and a validation F1 score of 0.2239579.

F1 Score: The F1 score is a metric that combines both precision and recall into a single value, providing a balanced measure of a model's performance. It is especially useful when dealing with imbalanced datasets

Training F1 Score: This is the F1 score calculated on the training dataset. It measures how well the model performs on the data it was trained on. A training F1 score of 0.2293934 means that the model achieved an F1 score of approximately 0.2294 when tested on the training data.

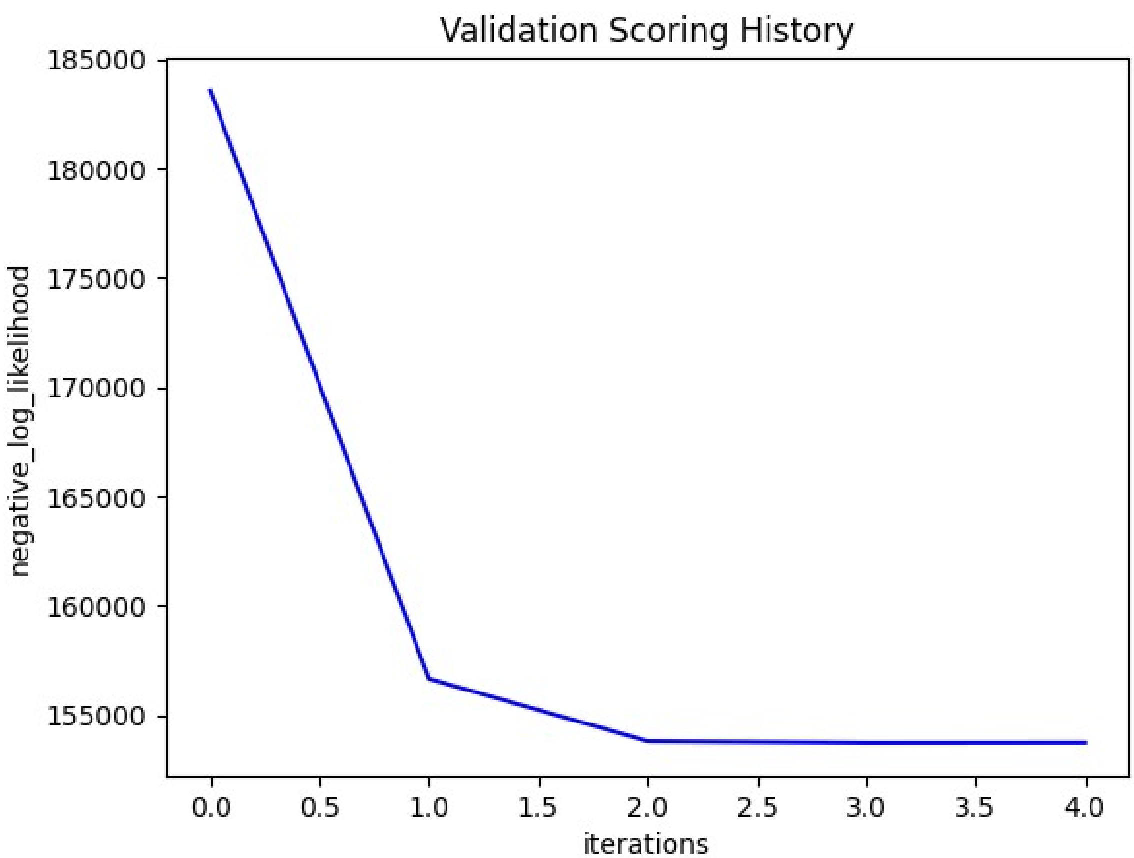
Validation F1 Score: This is the F1 score calculated on the validation dataset. The validation dataset is a separate set of data that the model did not see during training. The validation F1 score of 0.2239579 indicates that when the model is applied to new, unseen data (the validation dataset), it achieved an F1 score of approximately 0.2240. This demonstrates how well the model generalizes to data it has not been trained on.

Thresholds: In binary classification, different threshold values can be used to determine whether a prediction is classified as the positive or negative class. The explanation suggests that the F1 scores reported are associated with specific threshold values. Different thresholds can affect the trade-off between precision and recall, and they are used to fine-tune the model's performance.

```
In summary, the explanation indicates that a machine learning model
was evaluated using F1 scores on both the training and validation
datasets. The training F1 score represents its performance on the
training data, while the validation F1 score reflects how well the
model generalizes to new, unseen data. The thresholds mentioned
suggest that different threshold values were applied to calculate
these scores, influencing the precision and recall trade-off.
```

# Plot the Scoring history for any of our models, as shown below:

```
glm.plot(metric='negative_log_likelihood')
```



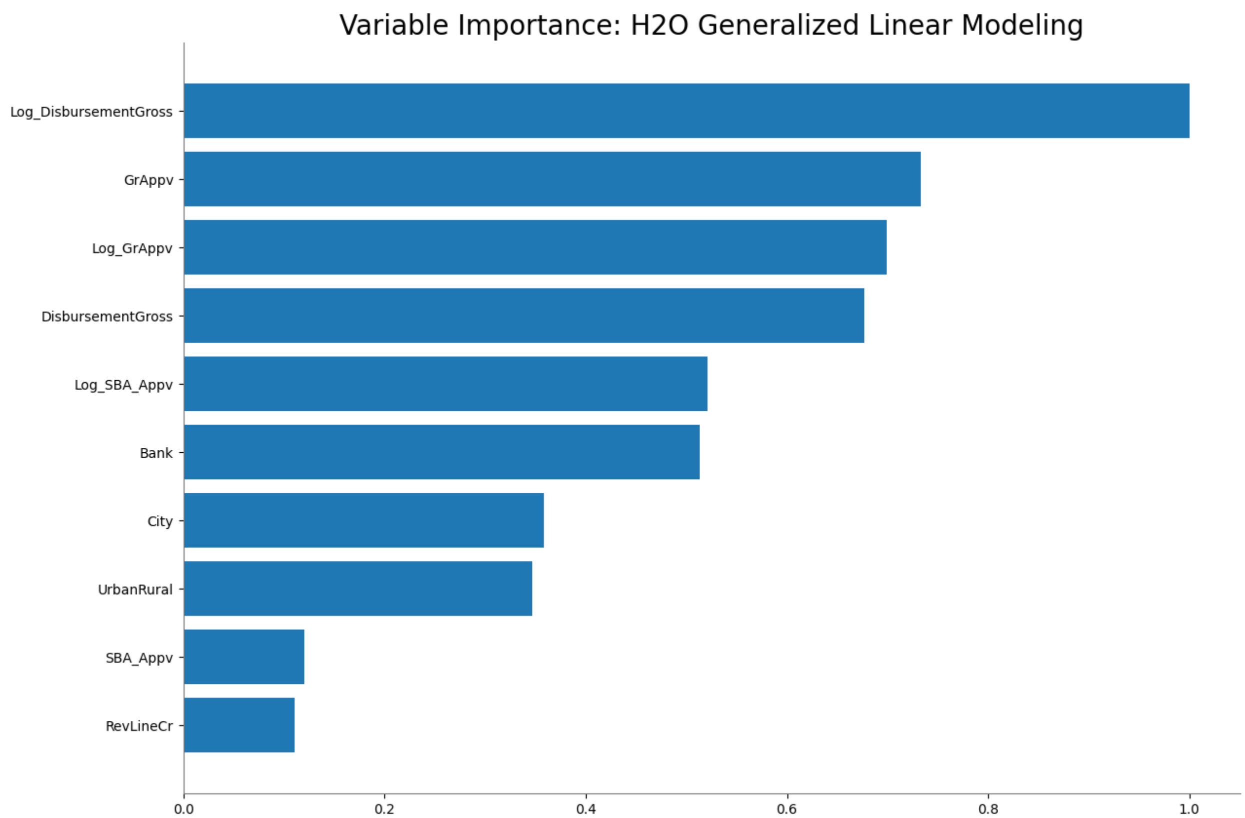

```
<h2o.plot._plot_result._MObject at 0x24155b7a260>
```

We can also generate a variable importance plot to see how each of our features contribute to the linear model.

```
As we can see after 2 iterations, the scores dont really improve after
this time We can also use the default number of iterations and use
early stopping; that way, the model will stop training when it is no
```

longer improving. We will use early stopping when we start tuning our models.

```
glm.varimp_plot()
```




```
<h2o.plot._plot_result._MObject at 0x241a2eeb430>

<Figure size 640x480 with 0 Axes>
```

From the variable importance plot, we can see that the most significant feature is Log_DisbursementGross. We can also see Gr_Appv, (Log_Gr_Appv) DisbursementGross, and Log_SBA_Appv are the next most important variables. As this is understood by the fact, that if your

DisbursementGross provides information about the size of the loans granted to small businesses. This information is crucial for understanding the financial impact of SBA loans on the businesses they support. The loan amount disbursed is often indicative of the level of risk associated with a borrower. Larger loan amounts may indicate higher financial stability, while smaller loans may be associated with smaller or riskier businesses.

SBA_Appv (Log_SBA_Appv) The variable confirms that the SBA has approved a loan for a specific borrower. It signifies that the borrower has successfully gone through the SBA's application and approval process Lenders and borrowers use this variable to determine if they are eligible for SBA loans and to understand the maximum loan amount that can be approved for their business.he variable is important for assessing the economic impact of SBA loans. By analyzing the approved loan amounts, one can estimate the potential economic impact in terms of job creation, business expansion, and overall economic growth.

he "Bank" variable helps identify the specific financial institutions that are participating in the SBA loan program. This information is crucial for understanding which banks are actively providing SBA loans to small businesses.

The Top 4 Variables with their relative importance is as follows:

Log_DisbursementGross: 1.0742711

Log_Gr_Appv: 0.7507896

Log_SBA_Appv: 0.5599474

Bank: 0.5510707

```
glm.predict(valid).head(10)
```

```
glm prediction progress: |

███████████████████████████████████████████████████████| (done) 100%

  predict        p0         p1
---------  --------  ---------
        0  0.970777  0.029223
        1  0.471275  0.528725
        0  0.951736  0.0482639
        0  0.951392  0.0486081
        0  0.945459  0.0545409
        0  0.92252   0.0774796
        0  0.855034  0.144966
        0  0.865822  0.134178
        0  0.919617  0.0803829
        1  0.759493  0.240507
[10 rows x 3 columns]
```

These columns contain the predicted probabilities for each class. "p0" represents the probability of an observation belonging to class 0 (Not Defaulted), and "p1" represents the probability of it belonging to class 1 (Defaulted on Loan). These probabilities can be used to assess the model's confidence in its predictions.

# H2O Model Tuning

`We Tune our model with lambda_search = True, as this will automatically tune the model. Other parameters that we can alter are max_active_predictors (feature selection parameter), nlambdas, which allows you to specify the number of lambda values, or the regularization strengths, to be used in the elastic net regularization path, and solver, which specify the algorithm or optimization method that the GLM model should use to find the solution

A value of alpha = 1 represents Lasso Regularization and a value of alpha = 0 produces Ridge regression

lambda is employed for regularization strength missing_value_handling parameter allows to specify how we want to handle any missing data (options are skip and MeanImputation)

```
glm_grid = h2o.grid.H2OGridSearch (

    H2OGeneralizedLinearEstimator(family = "binomial",
                                  lambda_search = True),

    hyper_params = {"alpha": [x*0.01 for x in range(0, 50)],
                    "missing_values_handling" : ["Skip",
"MeanImputation"]},

    grid_id = "glm_random_grid",

    search_criteria = {
        "strategy":"RandomDiscrete",
        "max_models":300,
        "max_runtime_secs":300,
        "seed":42})

%time glm_grid.train(x = train_X, y = train_y, training_frame = train,
validation_frame = valid)

glm Grid Build progress: |

███████████████████████████████████████████████████████| (done) 100%
CPU times: total: 1.61 s
Wall time: 5min 2s

Hyper-Parameter Search Summary: ordered by increasing logloss
     alpha                missing_values_handling    model_ids
logloss
---  -------------------  -------------------------
-----------------------  -------------------
     0.0                  Skip
glm_random_grid_model_11  0.39160808622118365
```

```
     0.0                    MeanImputation
glm_random_grid_model_34  0.3916080862218365
     0.44                   MeanImputation
glm_random_grid_model_28  0.391756991015515
     0.48                   Skip
glm_random_grid_model_2   0.3917586027842902
     0.48                   MeanImputation
glm_random_grid_model_36  0.3917586027842902
     0.43                   MeanImputation
glm_random_grid_model_10  0.39176212599720756
     0.43                   Skip
glm_random_grid_model_22  0.39176212599720756
     0.47000000000000003  MeanImputation
glm_random_grid_model_26  0.3917631576492127
     0.37                   Skip
glm_random_grid_model_13  0.3917744343513941
     0.25                   Skip
glm_random_grid_model_19  0.3917771796917925
---  ---                    ---                        ---
---
     0.07                   MeanImputation
glm_random_grid_model_20  0.39183529584589244
     0.07                   Skip
glm_random_grid_model_21  0.39183529584589244
     0.11                   MeanImputation
glm_random_grid_model_25  0.3918400598420588
     0.1                    MeanImputation
glm_random_grid_model_24  0.39184069991759385
     0.08                   Skip
glm_random_grid_model_23  0.39185013254495743
     0.06                   MeanImputation
glm_random_grid_model_9   0.3918788835170476
     0.05                   MeanImputation
glm_random_grid_model_4   0.3919370150415301
     0.03                   Skip
glm_random_grid_model_30  0.39213545399292443
     0.02                   Skip
glm_random_grid_model_27  0.3923318174271338
     0.29                   Skip
glm_random_grid_model_37  0.39418362265362783
[37 rows x 5 columns]
```

```
sorted_glm_grid = glm_grid.get_grid(sort_by = 'aucpr', decreasing =
True)
sorted_glm_grid.sorted_metric_table()
best_model_id = sorted_glm_grid.sorted_metric_table()['model_ids'][0]
best_model_id
```

```
'glm_random_grid_model_11'
```

```
he grid search results sorted by the AUC-PR metric in decreasing
order. AUC-PR (Area Under the Precision-Recall Curve) is a metric
commonly used to evaluate the performance of binary classification
models, especially when dealing with imbalanced datasets. Sorting in
decreasing order means that the models with the highest AUC-PR values
will appear at the top of the sorted list. Higher AUC-PR values
indicate better precision-recall trade-offs in the models.
```

sorted_metric_table() function provides an easy way to examine and analyze the results, allowing you to identify the best-performing models based on the chosen metric (AUC-PR in this case).

As in this case we can see the best model is `glm_random_grid_model_11` with AUCPR values as `0.439303`

```
tuned_glm = sorted_glm_grid.models[0]
tuned_glm.summary()

GLM Model: summary
    family    link    regularization               lambda_search
number_of_predictors_total    number_of_active_predictors
number_of_iterations    training_frame
--  --------  ------  --------------------------
----------------------------------------------------------------------
-----  ---------------------------  ----------------------------
---------------------  ---------------
    binomial  logit   Ridge ( lambda = 1.252E-5 )  nlambda = 30,
lambda.max = 12.525, lambda.min = 1.252E-5, lambda.1se = -1.0  27
27                             54                      py_17_sid_b6e5
```

```
Here we receive the best model for grid search along with the
parameters fro the best model
```

Lets evaluate the model on the validation set

```
tuned_glm_perf = tuned_glm.model_performance(valid)
print("Default GLM AUCPR: %.4f \nTuned GLM AUCPR:%.4f" % (glm.aucpr(),
tuned_glm_perf.aucpr()))

Default GLM AUCPR: 0.4455
Tuned GLM AUCPR:0.4393
```

```
Which suggests are default GLM model having higher AUCPR generalizes
better than the tuned model
```

# As such we will do Scoring in Scikit-Learn Model which has a better threshold

Saving the H2o model in the Artifacts

```
# Define the directory path and the model file name separately
model_directory =
"D:/Work/Gre/UTD/Courses/Fall/MIS6341/Softwares/Python/ml-fall-2023/
Project1/artifacts_h2o"
model_filename = "glm_model_h2o.pkl"

# Get the H2O model by its ID
model_h2o = h2o.get_model(best_model_id)

# Construct the full model path
model_path = f"{model_directory}/{model_filename}"

# Create an artifacts dictionary and include the model file path
artifacts_dict = {}
artifacts_dict["h2o_model_path"] = model_path

# Save the H2O model to the specified file path
h2o.save_model(model_h2o, model_path)
```

```
'D:\\Work\\Gre\\UTD\\Courses\\Fall\\MIS6341\\Softwares\\Python\\ml-
fall-2023\\Project1\\artifacts_h2o\\glm_model_h2o.pkl\\
glm_random_grid_model_11'
```

```
def score_with_h2o_model(input_data):
    import h2o
    try:
      h2o.cluster().shutdown()
    except:
      pass
    from h2o.frame import H2OFrame
    h2o.init(max_mem_size = "4G", nthreads=16)
    try:
        # Load the saved H2O model
        model_path =
"D:/Work/Gre/UTD/Courses/Fall\MIS6341/Softwares/Python/ml-fall-2023/
Project1/artifacts_h2o/glm_model_h2o.pkl/glm_random_grid_model_11"
        loaded_model = h2o.load_model(model_path)

        # Convert input_data to an H2O frame
        input_h2o = h2o.H2OFrame(input_data)

        # Use the loaded model for scoring
        predictions = loaded_model.predict(input_h2o)
```

```python
        # Extract the predictions as a Pandas DataFrame
        predictions_df = predictions.as_data_frame()

        # Return the prediction results
        return predictions_df

    except Exception as e:
        return f"Error: {e}"

score_with_h2o_model(valid)
```

```
H2O session _sid_b6e5 closed.
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 25.361-b09, mixed mode)
  Starting server from D:\Work\Gre\UTD\Courses\Fall\MIS6341\Softwares\
Python\ml-fall-2023\Lib\site-packages\h2o\backend\bin\h2o.jar
  Ice root: C:\Users\Asus\AppData\Local\Temp\tmps874sy9h
  JVM stdout: C:\Users\Asus\AppData\Local\Temp\tmps874sy9h\
h2o_Asus_started_from_python.out
  JVM stderr: C:\Users\Asus\AppData\Local\Temp\tmps874sy9h\
h2o_Asus_started_from_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:       America/Chicago
H2O_data_parsing_timezone:  UTC
H2O_cluster_version:        3.42.0.3
H2O_cluster_version_age:    2 months and 14 days
H2O_cluster_name:           H2O_from_python_Asus_vzln0q
H2O_cluster_total_nodes:    1
H2O_cluster_free_memory:    3.548 Gb
H2O_cluster_total_cores:    16
H2O_cluster_allowed_cores:  16
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:             3.10.11 final
--------------------------  -----------------------------

'Error: Argument `python_obj` should be a None | list | tuple | dict |
numpy.ndarray | pandas.DataFrame | scipy.sparse.issparse, got H2OFrame
City      State    Zip        Bank     BankState    NAICS       NoEmp
NewExist    CreateJob    RetainedJob    FranchiseCode    UrbanRural
```

```
RevLineCr     LowDoc    DisbursementGross    BalanceGross     GrAppv
SBA_Appv    MIS_Status    Log_DisbursementGross    Log_GrAppv
Log_SBA_Appv    Log_BalanceGross    TotalJobs    IncomeToLoanRatio
EmployeesToLoanRatio    JobPerLoan    Gauren_SBA_Appv    DefaultRate\
n0.11465    0.184773  93001  0.0314465     0.218517   235910
0.600407     0.17044    -0.0353733     -0.0454543               1
0.0716743     0.15307   0.187063              0.358949     -0.00229552
0.394801    0.410973            0                 0.306712
0.332752        0.344279         -0.00229816   -0.0808276
0.873414                1.46094    -0.196674            0.960651
17.5096\n0.337588   0.197662  11225  0.272221      0.220168   621111
-0.140136     0.186978   -0.022536      -0.0326467              0
0.243491      0.15307   0.187063             -0.451437     -0.00229552
-0.429073   -0.497867            0                 -0.600454     -
0.560494       -0.68889          -0.00229816   -0.0551827
0.906743                0.281473    0.110838            0.861822
17.5096\n0.0615385  0.124634  54935  0.175694      0.117429   453220
-0.140136     0.186978   -0.0353733     -0.0454543              1
0.0716743     0.15307   0.0897581            -0.593428     -0.00229552
-0.573427   -0.547995            0                 -0.899994     -
0.851971       -0.794062         -0.00229816   -0.0808276
1.08291                 0.255725    0.147497            1.04641
17.5096\n0.185864   0.188249  77379  0.147425      0.139976   332996
0.788908     0.17044    -0.0353733     -0.0454543              1
0.0716743     0.15307   0.187063              2.14249      -0.00229552
2.20803     2.02814             0                 1.14502
1.16566         1.10795          -0.00229816   -0.0808276
1.05638                 0.388981   -0.039853            1.0887
17.5096\n0.275041   0.184773  90001  0                 0.218517   448120
0.654264     0.17044     0.0288134     -0.0454543              1
0.243491      0.15307   0.187063              3.9295       -0.00229552
4.24307     5.44119             0                 1.59524
1.65691         1.86271          -0.00229816   -0.0166409
0.722176                0.120243   -0.00305832          0.779804
17.5096\n0.189995   0.156142  64068  0.145683      0.126444   811192
-0.140136     0.17044    -0.0353733     -0.0454543              1
0.0716743     0.15307   0.187063             -0.143213     -0.00229552
-0.115719   -0.129536            0                 -0.154566     -
0.12298        -0.138729         -0.00229816   -0.0808276
1.10558                 1.08183     0.623976            0.893333
17.5096\n0.175011   0.124634  54455  0.116923      0.117429   518210
-0.140136     0.17044    -0.0353733     -0.0411851              0
0.187265      0.251569  0.187063             -0.464802     -0.00229552
-0.573427   -0.587225            0                 -0.625118     -
0.851971       -0.884854         -0.00229816   -0.0765584
0.791522                0.238641    0.130373            0.976502
17.5096\n0.117995   0.140225  98116  0.175694      0.159167   812113
-0.072814     0.17044    -0.0310942     -0.01557                0
0.243491      0.251569  0.187063             -0.610744     -0.00229552
```

```
-0.591031   -0.598123             0                 -0.943518     -
0.894116       -0.911609         -0.00229816   -0.0466641
1.0211                  0.121737    0.0780177            0.988143
17.5096\n0.115      0.12935    2745  0.0896552     0.140419   448130
-0.140136     0.17044    -0.0353733     -0.0411851               1
0.243491      0.251569  0.187063             -0.489532     -0.00229552
-0.467802   -0.521841             0                 -0.672428     -
0.63074        -0.737812         -0.00229816   -0.0765584
0.938087                0.268541    0.146708             0.896445
17.5096\n0.276347   0.224201  60657  0.175694      0.159167   812111
0.0214369    0.17044    -0.0353733     -0.0454543              1
0.243491      0.15307   0.187063             -0.181308     -0.00229552
-0.154448   -0.100549             0                 -0.200048     -
0.167766       -0.105971         -0.00229816   -0.0808276
1.80318                -0.213198    0.803859             1.53604
17.5096\n[84951 rows x 29 columns]\n'
```