Info 6210 Project Portfolio

Yuan Chai Hsiang-Hua Chen

Abstract

Hyperparameters are parameters that are specified prior to running machine learning algorithms that have a large effect on the predictive power of statistical models parameter of a prior distribution, the term which used to distinguish them from parameters of the model for the underlying system under analysis. Knowledge of the relative importance of a hyperparameter to an algorithm and its range of values is crucial to hyperparameter tuning and creating effective models.

The hyperparameter database allows users to visualize and understand how to choose hyperparameters that maximize the predictive power of their models. The hyperparameter database is created by running millions of hyperparameter values, calculating the individual conditional expectation of every hyperparameter on the quality of a model. The data science part need to generating models using H2O to find best hyperparameters.

Background

The data we collected and stored concerns predicting housing transaction price which contains values of cities, floors, unit area households counts and parking capacity, rooms, heat fuel, heat type and front door structure. We separated and grouped data into different entities and attributes and build the one-to-many connections between them, which presented the data in more structured and organized way and allows us to query data, sort data, and manipulate data in various ways for the future performance.

Dataset

The dataset is from the website https://www.kaggle.com/econdata/predciting-price-transaction#trainPrice.csv . Housing price always been a popular item that people expect to predict. Since it is critical for us to find out the factors that affecting transaction price. This data set covers different aspects of factors which influence the housing price, which requires the scientific and specific method to calculate the best result.

Normalization

1NF

For all of our tables, We check them one by one and eliminate all the redundant data to ensure there are no repeating groups. We divided Alpha and lambda attributes in GLM Hyperparameter table into atomic as alpha one to seven and lambda one to five. And divided hiddens into hidden one to three in Deep Learning model. We make sure there are no same values in each table.

2NF

We check all the tables that whether there are any functional dependencies on part of any candidate key and make sure there are no partial dependencies.

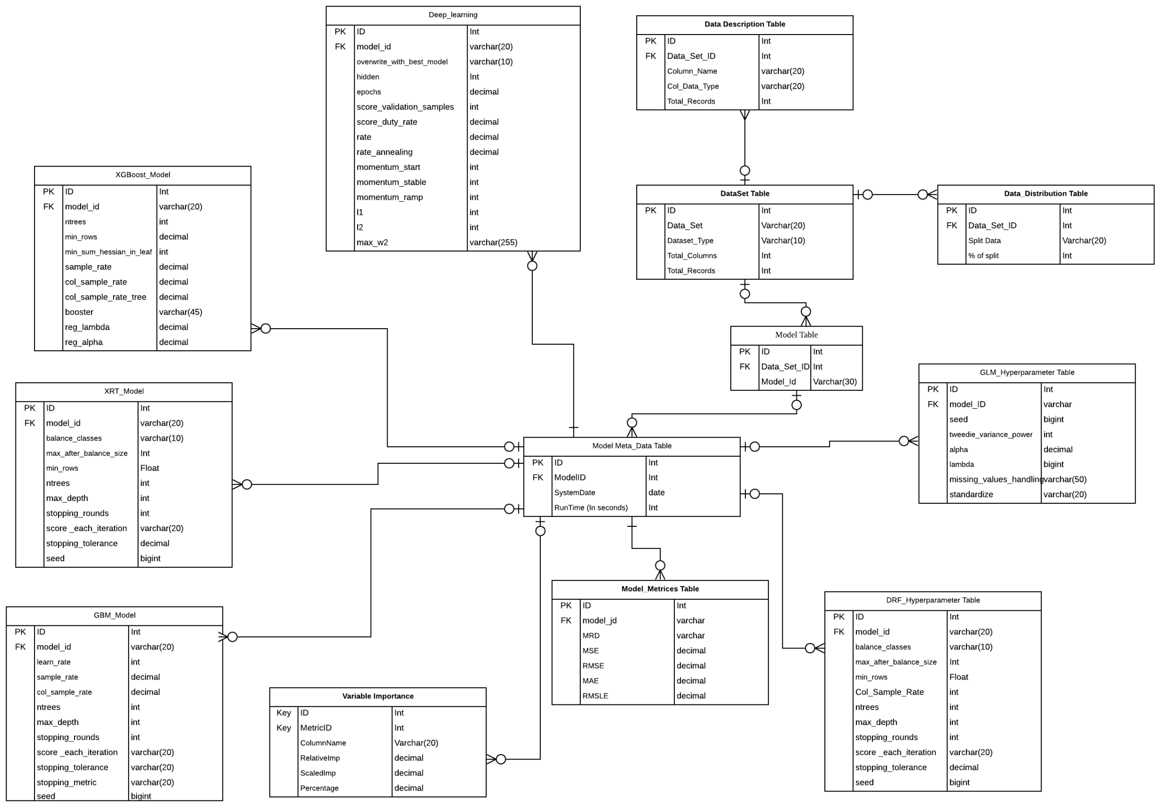
3NF

We check all our tables and make sure there are no non-prime attribute is transitively dependent of any key. All the fields are directly depend on the primary key.

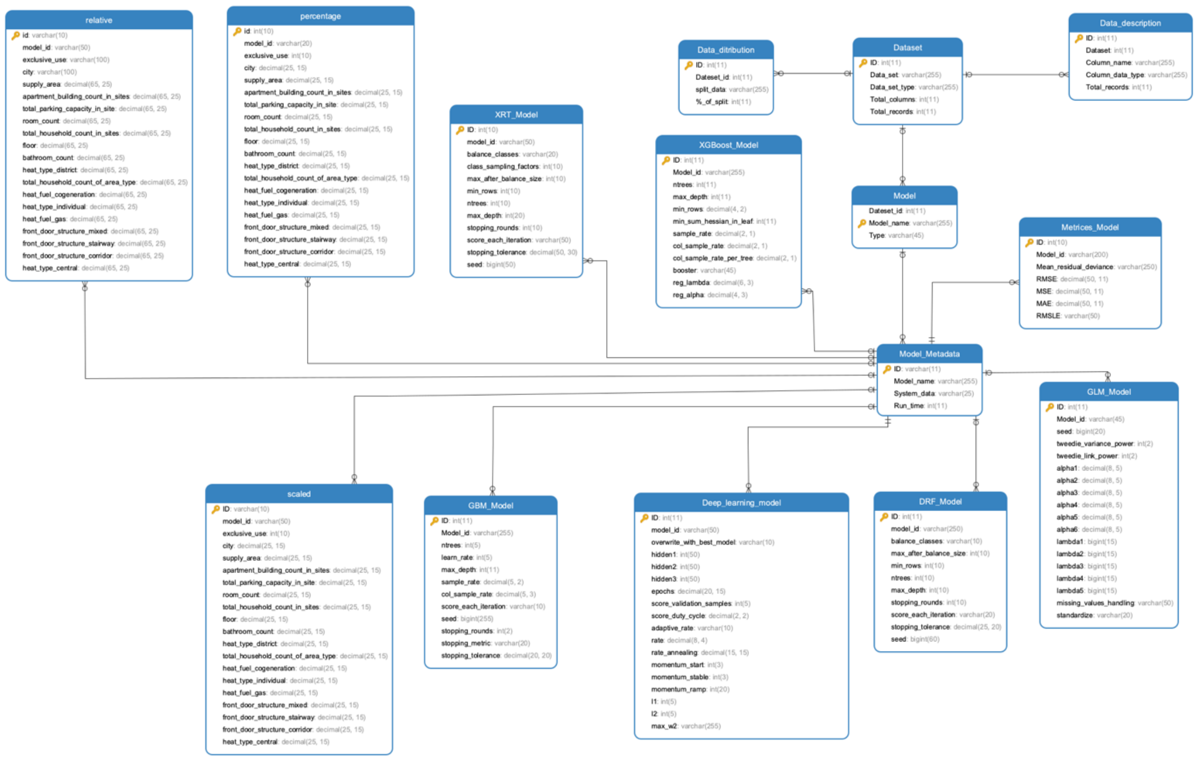
Info 6210 Project Portfolio

Yuan Chai Hsiang-Hua Chen

ERD



Physical Model



Info 6210 Project Portfolio

Yuan Chai   Hsiang-Hua Chen

10 Use Cases

1. Select the best model

```
SELECT me.Model_id, me.RMSE, mm.Model_name, mm.Run_time
FROM Metrices_Model me inner join Model_Metadata mm
ON me.Model_id    = mm.ID
order by RMSE desc
LIMIT 1;
```

Result Grid   Filter Rows: Export:

Model_id	RMSE	Model_name	Run_time
XGB_81	487402634.50000000000	XGBoost_grid_1_AutoML_20190421_193027_...	300

2. Select the average rmse with the same type model

```
SELECT avg(mm.RMSE) as GLM_AVERAGE_rmse
FROM Metrices_Model mm
WHERE mm.Model_id LIKE "G%";
```

GLM_AVERAGE_rmse
200646054.010858143607706

3. Select the counts of models which runtime is 2000

```
SELECT COUNT(*)
FROM Model_Metadata mm
WHERE mm.Run_time = 2000
order BY mm.Run_time;
```

Result Grid   Filter Rows:

COUNT(*)
390

4. Select the hyperparameter of the same deep learning model but different runtime.

```
SELECT mm.ID, mm.Model_name, mm.Run_time, me.RMSE,
dl. hidden1, hidden2,hidden3,epochs
FROM Model_Metadata mm left join Deep_learning_model dl
ON mm.ID = dl.model_id
JOIN Metrices_Model me
ON    dl.model_id = me.Model_id
WHERE mm.ID LIKE "DL%"
ORDER BY mm.Model_name, me.RMSE, mm.Run_time desc;
```

Result Grid Filter Rows: Search Export:

ID	Model_name	Run_time	RMSE	hidden1	hidden2	hidden3	epochs
DL_5	DeepLearning_1_AutoML_20190421_193027	2000	190896330.40000000000	10	10	10	10.423572230000000
DL_4	DeepLearning_1_AutoML_20190421_193027	1500	190896330.40000000000	10	10	10	10.423572230000000
DL_3	DeepLearning_1_AutoML_20190421_193027	1000	190896330.40000000000	10	10	10	10.423572230000000
DL_2	DeepLearning_1_AutoML_20190421_193027	500	190896330.40000000000	10	10	10	10.423572230000000
DL_1	DeepLearning_1_AutoML_20190421_193027	300	190896330.40000000000	10	10	10	10.423572230000000
DL_9	DeepLearning_1_AutoML_20190421_193852	2000	192326606.10000000000	10	10	10	10.406606940000000
DL_8	DeepLearning_1_AutoML_20190421_193852	1500	192326606.10000000000	10	10	10	10.406606940000000
DL_7	DeepLearning_1_AutoML_20190421_193852	1000	192326606.10000000000	10	10	10	10.406606940000000
DL_6	DeepLearning_1_AutoML_20190421_193852	500	192326606.10000000000	10	10	10	10.406606940000000
DL_12	DeepLearning_1_AutoML_20190421_195113	2000	191927389.20000000000	10	10	10	10.405991040000000
DL_11	DeepLearning_1_AutoML_20190421_195113	1500	191927389.20000000000	10	10	10	10.405991040000000
DL_10	DeepLearning_1_AutoML_20190421_195113	1000	191927389.20000000000	10	10	10	10.405991040000000
DL_14	DeepLearning_1_AutoML_20190421_201138	2000	195714904.40000000000	10	10	10	10.394288910000000
DL_13	DeepLearning_1_AutoML_20190421_201138	1500	195714904.40000000000	10	10	10	10.394288910000000
DL_15	DeepLearning_1_AutoML_20190421_204058	2000	191215497.60000000000	10	10	10	10.405543110000000
DL_20	DeepLearning_grid_1_AutoML_20190421_193...	2000	204429500.10000000000	500	0	0	22.400000000000000
DL_19	DeepLearning_grid_1_AutoML_20190421_193...	1500	204429500.10000000000	500	0	0	22.400000000000000
DL_18	DeepLearning_grid_1_AutoML_20190421_193...	1000	204429500.10000000000	500	0	0	22.400000000000000
DL_17	DeepLearning_grid_1_AutoML_20190421_193...	500	204429500.10000000000	500	0	0	22.400000000000000
DL_16	DeepLearning_grid_1_AutoML_20190421_193...	300	204429500.10000000000	500	0	0	22.400000000000000
DL_25	DeepLearning_grid_1_AutoML_20190421_193...	2000	213497943.90000000000	200	0	0	8.000000000000000
DL_24	DeepLearning_grid_1_AutoML_20190421_193...	1500	213497943.90000000000	200	0	0	8.000000000000000
DL_23	DeepLearning_grid_1_AutoML_20190421_193...	1000	213497943.90000000000	200	0	0	8.000000000000000
DL_22	DeepLearning_grid_1_AutoML_20190421_193...	500	213497943.90000000000	200	0	0	8.000000000000000
DL_21	DeepLearning_grid_1_AutoML_20190421_193...	300	213497943.90000000000	200	0	0	8.000000000000000
DL_30	DeepLearning_grid_1_AutoML_20190421_193...	2000	198521257.60000000000	200	200	200	7.559182531000000
DL_29	DeepLearning_grid_1_AutoML_20190421_193...	1500	198521257.60000000000	200	200	0	7.559182531000000
DL_28	DeepLearning_grid_1_AutoML_20190421_193...	1000	198521257.60000000000	200	200	0	7.559182531000000
DL_27	DeepLearning_grid_1_AutoML_20190421_193...	500	198521257.60000000000	200	200	0	7.559182531000000
DL_26	DeepLearning_grid_1_AutoML_20190421_193...	300	198521257.60000000000	200	200	0	7.559182531000000
DL_34	DeepLearning_grid_1_AutoML_20190421_193...	2000	190149276.40000000000	500	0	0	62.400000000000000
DL_33	DeepLearning_grid_1_AutoML_20190421_193...	1500	190149276.40000000000	500	0	0	62.400000000000000
DL_32	DeepLearning_grid_1_AutoML_20190421_193...	1000	190149276.40000000000	500	0	0	62.400000000000000
DL_31	DeepLearning_grid_1_AutoML_20190421_193...	500	190149276.40000000000	500	0	0	62.400000000000000
DL_38	DeepLearning_grid_1_AutoML_20190421_193...	2000	196963462.60000000000	50	0	0	8.000000000000000
DL_37	DeepLearning_grid_1_AutoML_20190421_193...	1500	196963462.60000000000	50	0	0	8.000000000000000
DL_36	DeepLearning_grid_1_AutoML_20190421_193...	1000	196963462.60000000000	50	0	0	8.000000000000000
DL_35	DeepLearning_grid_1_AutoML_20190421_193...	500	196963462.60000000000	50	0	0	8.000000000000000

Result 5

5. Select ID, name, runtime and rmse which is higher than the average rmse of all XGB model

```
SELECT metr.ID, metr.Model_id,mm.Model_name, mm.Run_time, metr.RMSE
FROM Metrices_Model metr INNER JOIN Model_Metadata mm
ON metr.Model_id = mm.ID
WHERE metr.Model_id LIKE "XG%"
HAVING metr.RMSE > (
SELECT avg(metr.RMSE)
FROM Metrices_Model metr
WHERE metr.Model_id LIKE "XG%"
)
ORDER BY metr.RMSE DESC;
```

Result Grid Filter Rows: Search Export:

ID	Model_id	Model_name	Run_time	RMSE
773	XGB_81	XGBoost_grid_1_AutoML_20190421_193027_...	300	487402634.50000000000
774	XGB_82	XGBoost_grid_1_AutoML_20190421_193027_...	500	487402634.50000000000
775	XGB_83	XGBoost_grid_1_AutoML_20190421_193027_...	1000	487402634.50000000000
776	XGB_84	XGBoost_grid_1_AutoML_20190421_193027_...	1500	487402634.50000000000
777	XGB_85	XGBoost_grid_1_AutoML_20190421_193027_...	2000	487402634.50000000000
911	XGB_219	XGBoost_grid_1_AutoML_20190421_201138_...	1500	450945562.60000000000
912	XGB_220	XGBoost_grid_1_AutoML_20190421_201138_...	2000	450945562.60000000000
814	XGB_122	XGBoost_grid_1_AutoML_20190421_193852_...	500	432719232.60000000000
815	XGB_123	XGBoost_grid_1_AutoML_20190421_193852_...	1000	432719232.60000000000
816	XGB_124	XGBoost_grid_1_AutoML_20190421_193852_...	1500	432719232.60000000000
817	XGB_125	XGBoost_grid_1_AutoML_20190421_193852_...	2000	432719232.60000000000
927	XGB_235	XGBoost_grid_1_AutoML_20190421_204058_...	2000	398515351.40000000000
873	XGB_181	XGBoost_grid_1_AutoML_20190421_201138_...	1500	370935717.60000000000
874	XGB_182	XGBoost_grid_1_AutoML_20190421_201138_...	2000	370935717.60000000000
931	XGB_239	XGBoost_grid_1_AutoML_20190421_204058_...	2000	356043480.90000000000
758	XGB_66	XGBoost_grid_1_AutoML_20190421_193027_...	300	345639699.70000000000
759	XGB_67	XGBoost_grid_1_AutoML_20190421_193027_...	500	345639699.70000000000
760	XGB_68	XGBoost_grid_1_AutoML_20190421_193027_...	1000	345639699.70000000000
761	XGB_69	XGBoost_grid_1_AutoML_20190421_193027_...	1500	345639699.70000000000
762	XGB_70	XGBoost_grid_1_AutoML_20190421_193027_...	2000	345639699.70000000000
875	XGB_183	XGBoost_grid_1_AutoML_20190421_201138_...	1500	344434622.40000000000
876	XGB_184	XGBoost_grid_1_AutoML_20190421_201138_...	2000	344434622.40000000000
946	XGB_254	XGBoost_grid_1_AutoML_20190421_204058_...	2000	343505776.20000000000
881	XGB_189	XGBoost_grid_1_AutoML_20190421_201138_...	1500	340762997.70000000000
882	XGB_190	XGBoost_grid_1_AutoML_20190421_201138_...	2000	340762997.70000000000
806	XGB_114	XGBoost_grid_1_AutoML_20190421_193852_...	500	329104524.70000000000
807	XGB_115	XGBoost_grid_1_AutoML_20190421_193852_...	1000	329104524.70000000000
808	XGB_116	XGBoost_grid_1_AutoML_20190421_193852_...	1500	329104524.70000000000
809	XGB_117	XGBoost_grid_1_AutoML_20190421_193852_...	2000	329104524.70000000000

Result 7

6. Select the top 10 rmse in XRT model

```
SELECT metr.ID, metr.Model_id,mm.Model_name, mm.Run_time, metr.RMSE
FROM Metrices_Model metr INNER JOIN Model_Metadata mm
ON metr.Model_id = mm.ID
WHERE metr.Model_id LIKE "XRT%"
ORDER BY metr.RMSE
LIMIT 10;
```

Result Grid Filter Rows: Search Export:

ID	Model_id	Model_name	Run_time	RMSE
962	XRT_10	XRT_1_AutoML_20190421_195113	1000	163324521.90000000000
963	XRT_11	XRT_1_AutoML_20190421_195113	1500	163324521.90000000000
964	XRT_12	XRT_1_AutoML_20190421_195113	2000	163324521.90000000000
958	XRT_6	XRT_1_AutoML_20190421_193852	500	163709370.90000000000
959	XRT_7	XRT_1_AutoML_20190421_193852	1000	163709370.90000000000
960	XRT_8	XRT_1_AutoML_20190421_193852	1500	163709370.90000000000
961	XRT_9	XRT_1_AutoML_20190421_193852	2000	163709370.90000000000
967	XRT_15	XRT_1_AutoML_20190421_204058	2000	164549230.30000000000
965	XRT_13	XRT_1_AutoML_20190421_201138	1500	167114919.80000000000
966	XRT_14	XRT_1_AutoML_20190421_201138	2000	167114919.80000000000

7. Select the range of the learning rate of all the model

```
SELECT
MIN(gm.ntrees) AS min_ntrees,
MAX(gm.ntrees) AS max_ntrees,
MIN(gm.max_depth) AS min_max_depth,
MAX(gm.max_depth) AS max_max_depth
FROM GBM_Model gm;
```

Result Grid Filter Rows: Search Export:

min_ntrees	max_ntrees	min_max_depth	max_max_depth
1	487	3	17

8. Select all cities which variable is higher than 0.08

```
SELECT *
FROM percentage pp
having pp.city > 0.08
order by pp.city desc;
```

Result Grid   Filter Rows: Search   Edit: Export/Import:

id	model_id	exclusive_use	city	supply_area	apartment_building_count_in_sites	total_parking_capacity_in_site	room_count
505	GBM_384	0	0.272907000000000	0.215423000000000	0.055961000000000	0.032453000000000	0.027871000000000
506	GBM_385	0	0.272907000000000	0.215423000000000	0.055961000000000	0.032453000000000	0.027871000000000
806	XGB_114	0	0.272324000000000	0.249439000000000	0.058628000000000	0.036207000000000	0.028727000000000
807	XGB_115	0	0.272324000000000	0.249439000000000	0.058628000000000	0.036207000000000	0.028727000000000
808	XGB_116	0	0.272324000000000	0.249439000000000	0.058628000000000	0.036207000000000	0.028727000000000
809	XGB_117	0	0.272324000000000	0.249439000000000	0.058628000000000	0.036207000000000	0.028727000000000
622	GBM_501	0	0.269426000000000	0.178430000000000	0.043559000000000	0.032996000000000	0.032795000000000
935	XGB_243	0	0.265433000000000	0.208181000000000	0.044047000000000	0.037861000000000	0.031473000000000
649	GBM_528	0	0.265077000000000	0.190390000000000	0.051549000000000	0.049223000000000	0.034133000000000
881	XGB_189	0	0.264005000000000	0.198287000000000	0.055712000000000	0.040466000000000	0.032623000000000
882	XGB_190	0	0.264005000000000	0.198287000000000	0.055712000000000	0.040466000000000	0.032623000000000
945	XGB_253	0	0.261534000000000	0.234494000000000	0.057820000000000	0.040411000000000	0.028857000000000
778	XGB_86	0	0.260691000000000	0.172694000000000	0.046440000000000	0.041636000000000	0.036042000000000
779	XGB_87	0	0.260691000000000	0.172694000000000	0.046440000000000	0.041636000000000	0.036042000000000
780	XGB_88	0	0.260691000000000	0.172694000000000	0.046440000000000	0.041636000000000	0.036042000000000
781	XGB_89	0	0.260691000000000	0.172694000000000	0.046440000000000	0.041636000000000	0.036042000000000
478	GBM_357	0	0.259891000000000	0.157335000000000	0.050411000000000	0.036460000000000	0.033786000000000
479	GBM_358	0	0.259891000000000	0.157335000000000	0.050411000000000	0.036460000000000	0.033786000000000
480	GBM_359	0	0.259891000000000	0.157335000000000	0.050411000000000	0.036460000000000	0.033786000000000
439	GBM_318	0	0.258275000000000	0.183292000000000	0.040692000000000	0.033031000000000	0.030988000000000
440	GBM_319	0	0.258275000000000	0.183292000000000	0.040692000000000	0.033031000000000	0.030988000000000
441	GBM_320	0	0.258275000000000	0.183292000000000	0.040692000000000	0.033031000000000	0.030988000000000
599	GBM_478	0	0.255531000000000	0.174088000000000	0.044037000000000	0.036313000000000	0.030672000000000
600	GBM_479	0	0.255531000000000	0.174088000000000	0.044037000000000	0.036313000000000	0.030672000000000
593	GBM_472	0	0.255442000000000	0.171609000000000	0.030863000000000	0.030723000000000	0.029318000000000
594	GBM_473	0	0.255442000000000	0.171609000000000	0.030863000000000	0.030723000000000	0.029318000000000

percentage 12

9. Select all the runtime from the dataset

```
SELECT distinct mm.Run_time
from Model_Metadata mm
order by mm.Run_time;
```

Result Grid

Run_time
300
500
1000
1500
2000

10. Select the type, counts of the type and the best RMSE by order from model data tables

```
SELECT mo.TYPE AS MODEL_TYPE, COUNT(*) AS AMOUNT,
          MAX(me.RMSE) AS BEST_RMSE
FROM Model_Metadata mm,   Model mo, Metrices_Model me
WHERE mm.Model_name = mo.Model_name AND   mm.ID = me.Model_id
GROUP BY mo.type
ORDER BY max(me.RMSE) desc ;
```

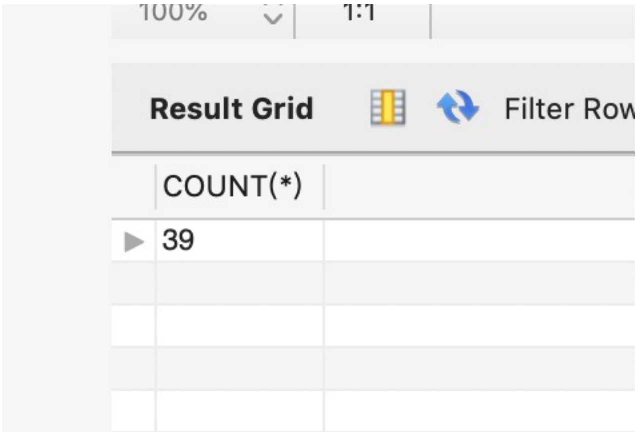
Result Grid Filter Rows: Search

MODEL_TYPE	AMOUNT	BEST_RMSE
XGB	260	487402634.50000000000
GLM	15	317000684.00000000000
GBM	556	316428928.70000000000
DL	106	244580272.80000000000
DRF	15	168065773.30000000000
XRT	15	167131676.80000000000

4 Views

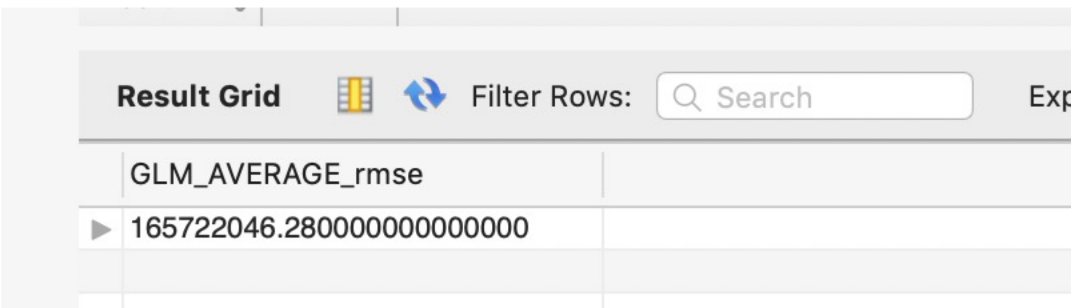
1. Select all the counts of deep learning models in model

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `finall`.`case1` AS
    SELECT
        COUNT(0) AS `COUNT(*)`
    FROM
        `finall`.`model` `m`
    WHERE
        (`m`.`Type` = 'DL')
```



2. Select average RMSE in DRF model

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `finall`.`case4` AS
    SELECT
        AVG(`mm`.`RMSE`) AS `GLM_AVERAGE_rmse`
    FROM
        `finall`.`metrices_model` `mm`
    WHERE
        (`mm`.`Model_id` LIKE 'DRF%')
```



3. Select GLM ID, model name and standardize from metadata by runtime order

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `finall`.`case2` AS
    SELECT
        `mm`.`ID` AS `ID`,
        `mm`.`Model_name` AS `Model_name`,
        `dl`.`standardize` AS `standardize`
    FROM
        (`finall`.`model_metadata` `mm`
        JOIN `finall`.`glm_model` `dl` ON ((`mm`.`ID` = `dl`.`Model_id`)))
    ORDER BY `mm`.`Model_name` , `mm`.`Run_time` DESC
```

Result Grid Filter Rows: Search Export:

ID	Model_name	standardize
GLM_5	GLM_grid_1_AutoML_20190421_193027_mod...	TRUE
GLM_4	GLM_grid_1_AutoML_20190421_193027_mod...	TRUE
GLM_3	GLM_grid_1_AutoML_20190421_193027_mod...	TRUE
GLM_2	GLM_grid_1_AutoML_20190421_193027_mod...	TRUE
GLM_1	GLM_grid_1_AutoML_20190421_193027_mod...	TRUE
GLM_9	GLM_grid_1_AutoML_20190421_193852_mod...	TRUE
GLM_8	GLM_grid_1_AutoML_20190421_193852_mod...	TRUE
GLM_7	GLM_grid_1_AutoML_20190421_193852_mod...	TRUE
GLM_6	GLM_grid_1_AutoML_20190421_193852_mod...	TRUE
GLM_12	GLM_grid_1_AutoML_20190421_195113_mod...	TRUE
GLM_11	GLM_grid_1_AutoML_20190421_195113_mod...	TRUE
GLM_10	GLM_grid_1_AutoML_20190421_195113_mod...	TRUE
GLM_14	GLM_grid_1_AutoML_20190421_201138_mod...	TRUE
GLM_13	GLM_grid_1_AutoML_20190421_201138_mod...	TRUE
GLM_15	GLM_grid_1_AutoML_20190421_204058_mod...	TRUE

4. Select top 20 DL MSE and runtime in metrices model

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `finall`.`case3` AS
    SELECT
        `metr`.`ID` AS `ID`,
        `metr`.`Model_id` AS `Model_id`,
        `mm`.`Model_name` AS `Model_name`,
        `mm`.`Run_time` AS `Run_time`,
        `metr`.`MSE` AS `MSE`
    FROM
        (`finall`.`metrices_model` `metr`
        JOIN `finall`.`model_metadata` `mm` ON ((`metr`.`Model_id` = `mm`.`ID`)))
    WHERE
        (`metr`.`Model_id` LIKE 'DL%')
    ORDER BY `metr`.`MSE`
    LIMIT 20
```

Result Grid Filter Rows: Search Export:

ID	Model_id	Model_name	Run_time	MSE
98	DL_98	DeepLearning_grid_1_AutoML_20190421_204...	2000	318000000000000000.00000000000
90	DL_90	DeepLearning_grid_1_AutoML_20190421_2011...	1500	323000000000000000.00000000000
91	DL_91	DeepLearning_grid_1_AutoML_20190421_2011...	2000	323000000000000000.00000000000
86	DL_86	DeepLearning_grid_1_AutoML_20190421_2011...	1500	328000000000000000.00000000000
87	DL_87	DeepLearning_grid_1_AutoML_20190421_2011...	2000	328000000000000000.00000000000
101	DL_101	DeepLearning_grid_1_AutoML_20190421_204...	2000	332000000000000000.00000000000
99	DL_99	DeepLearning_grid_1_AutoML_20190421_204...	2000	332000000000000000.00000000000
63	DL_63	DeepLearning_grid_1_AutoML_20190421_1951...	1000	335000000000000000.00000000000
64	DL_64	DeepLearning_grid_1_AutoML_20190421_1951...	1500	335000000000000000.00000000000
65	DL_65	DeepLearning_grid_1_AutoML_20190421_1951...	2000	335000000000000000.00000000000
72	DL_72	DeepLearning_grid_1_AutoML_20190421_1951...	1000	344000000000000000.00000000000
73	DL_73	DeepLearning_grid_1_AutoML_20190421_1951...	1500	344000000000000000.00000000000
74	DL_74	DeepLearning_grid_1_AutoML_20190421_1951...	2000	344000000000000000.00000000000
84	DL_84	DeepLearning_grid_1_AutoML_20190421_2011...	1500	347000000000000000.00000000000
85	DL_85	DeepLearning_grid_1_AutoML_20190421_2011...	2000	347000000000000000.00000000000
92	DL_92	DeepLearning_grid_1_AutoML_20190421_2011...	1500	353000000000000000.00000000000
93	DL_93	DeepLearning_grid_1_AutoML_20190421_2011...	2000	353000000000000000.00000000000
31	DL_31	DeepLearning_grid_1_AutoML_20190421_193...	500	362000000000000000.00000000000
32	DL_32	DeepLearning_grid_1_AutoML_20190421_193...	1000	362000000000000000.00000000000
33	DL_33	DeepLearning_grid_1_AutoML_20190421_193...	1500	362000000000000000.00000000000

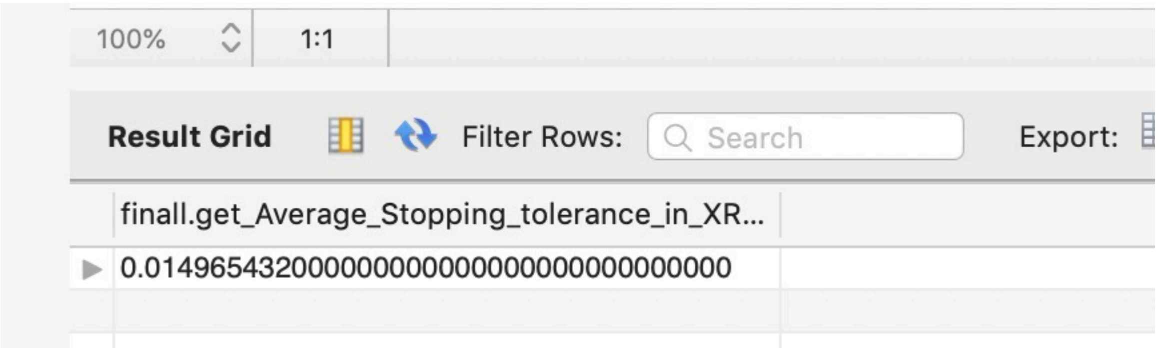
Info 6210 Project Portfolio

Yuan Chai Hsiang-Hua Chen

4 Functions

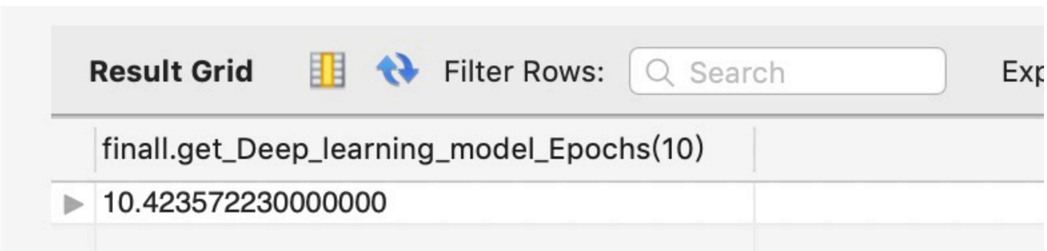
1. Get_Average_Stopping_tolerance_in_XRT_Model

```
CREATE DEFINER=`root`@`localhost`
FUNCTION `get_Average_Stopping_tolerance_in_XRT_Model`(id int)
RETURNS varchar(500) CHARSET utf8
BEGIN
   DECLARE a varchar(500);
   SELECT AVG(stopping_tolerance) INTO a FROM XRT_Model
   WHERE ID = id;
RETURN (a);
END
```



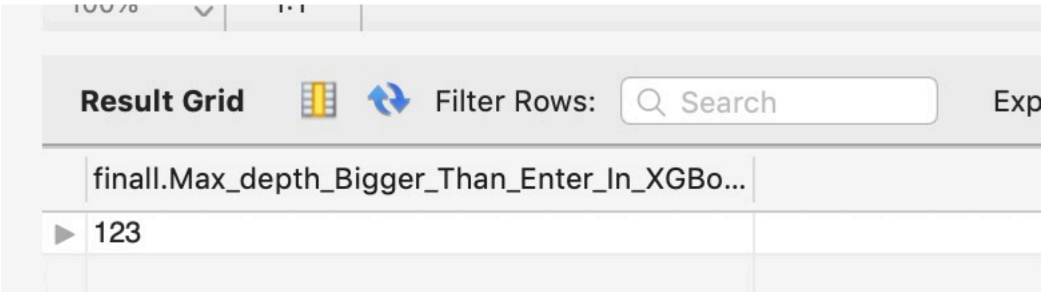
2. get_Deep_learning_model_Epochs

```
CREATE DEFINER=`root`@`localhost`
FUNCTION `get_Deep_learning_model_Epochs`(id int) RETURNS varchar(500) CHARSET utf8
BEGIN
   DECLARE a varchar(500);
   SELECT epochs INTO a FROM Deep_learning_model
   WHERE ID = id limit 1;
RETURN (a);
END
```



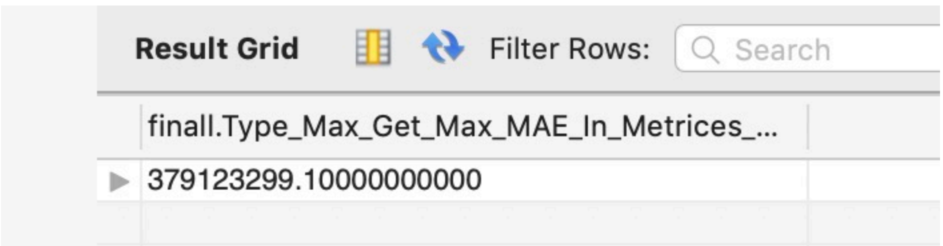
3. Get Max_depth_Bigger_Than_Enter_In_XGBoost_Model

```
CREATE DEFINER=`root`@`localhost`
FUNCTION `Max_depth_Bigger_Than_Enter_In_XGBoost_Model`(EnteredNum int)
RETURNS varchar(50) CHARSET utf8
BEGIN
 DECLARE b BIGINT;
 SELECT count(max_depth) INTO b FROM XGBoost_Model
 WHERE max_depth > EnteredNum;
RETURN (b);
END
```



4. Type_Max_Get_Max_MAE_In_Metrices_Model

```
CREATE DEFINER=`root`@`localhost`
FUNCTION `Type_Max_Get_Max_MAE_In_Metrices_Model`(Enter VARCHAR(50))
RETURNS varchar(500) CHARSET utf8
BEGIN
   DECLARE namemodel VARCHAR(500);
   SELECT max(MAE) INTO namemodel FROM Metrices_Model
   where maxx = "max";
   RETURN namemodel;
END
```



Analytics & Conclusions

By storing Hyperparameters data set in the database enables us obtain the structural and organized data to call the different functions for analyzing and select the best model for prediction, which make it more visualized to check and use the data and achieve different utilization. By creating the use cases, functions and views, we can select single or combined date set, get the best model, calculate the average or the max data for improving the different performance.

Contributions

- Hsiang-Hua Chen : 20% from the website, 80% created by myself

- Yuan Chai : 25% from the website, 75% created by myself

Citations

https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/
https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)
https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a
https://towardsdatascience.com/hyperparameters-in-deep-learning-927f7b2084dd
https://www.w3schools.com/sql/sql_create_index.asp
https://docs.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-2017
https://www.w3schools.com/sql/sql_view.asp

License