Hyperparameter project
Quanhan Sun/Guoyan Li
DB Group 06

1.Abstract:

In hyperparameter database, our objective is to analyze the effect of hyperparameters on the following algorithms: Distributed random forest, generalized linear model, gradient boosting machine, naïve Bayes classifier and so on.

The hyperparameter database also uses these data to build models that can predict hyperparameters without search and for visualizing and teach statistical concepts such as power and bias/variance tradeoff.

2. Data source:

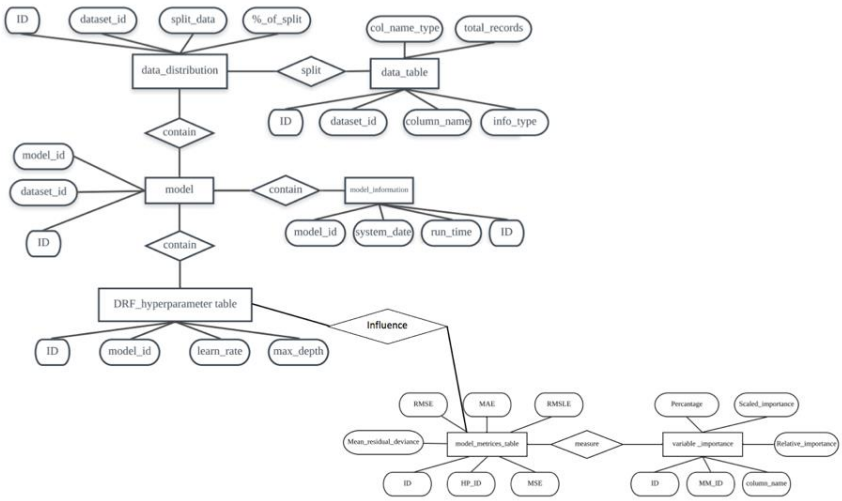
Our data source: https://www.kaggle.com/c/house-prices-advanced-regression-techniques, our training dataset has 81 columns, with 1460 records and our text dataset has 80 columns and with 1459 records.

3.Background:

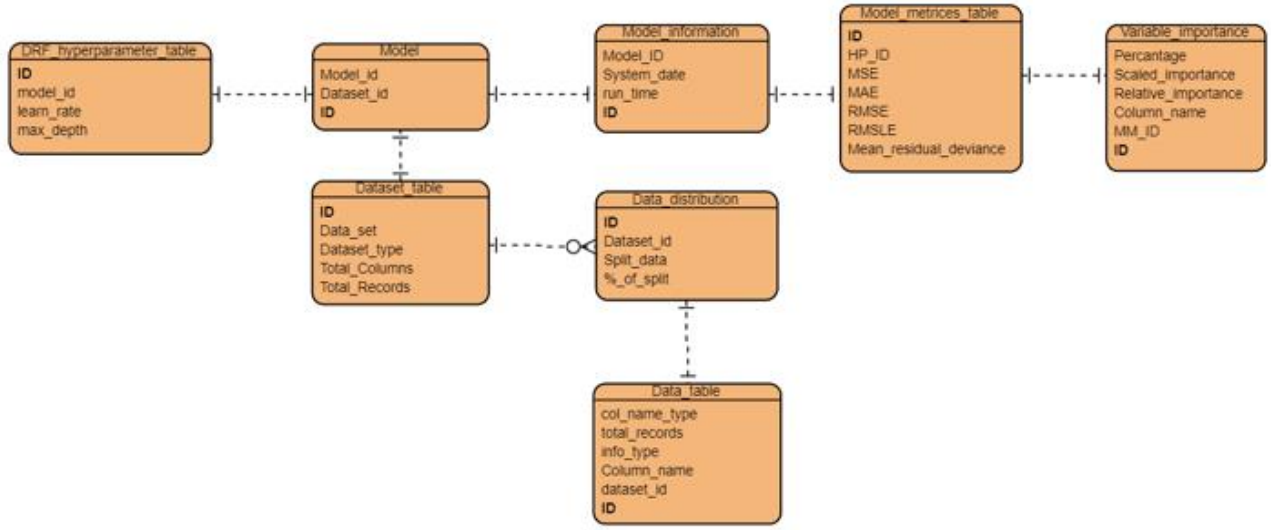
The data we collected and stored concerns predicting housing transaction price which contains values of cities, floors, unit area households counts and parking capacity, rooms, heat fuel, heat type and front door structure. We separated and grouped data into different entities and attributes and build 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.

3. Conceptual model:

Our conceptual model has seven tables and we can clearly find the relationship between each table.

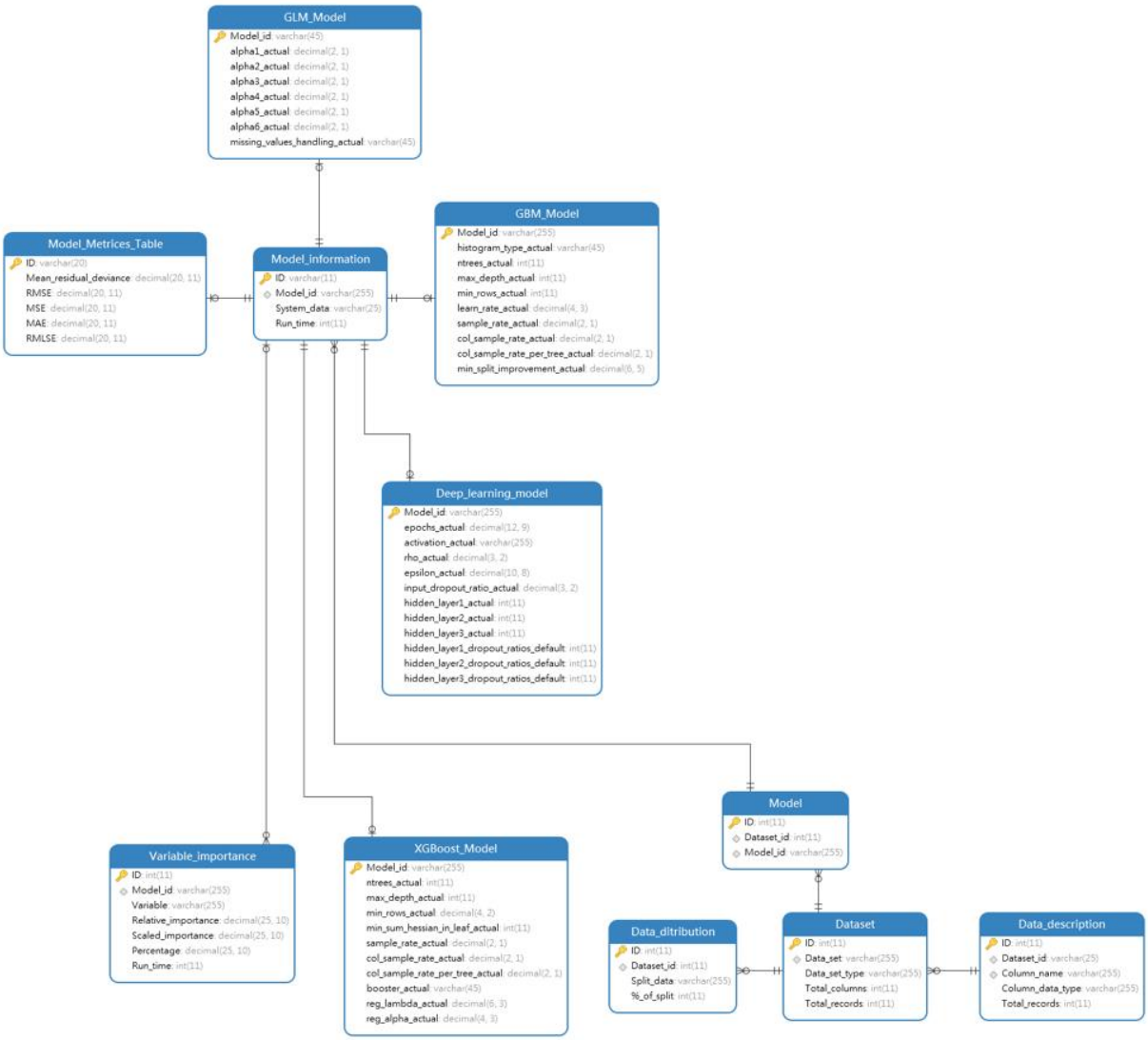


4. E-R diagram:



this our E-R diagram before we create the physical database.

5.Physical model:



6.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.

7. 10 use cases:

7.1 Find the best model name and corresponding RMSE value which evaluate the mod el performance.

```
SELECT distinct a.ID ,a.Model_id,b.RMSE
FROM Model_information a, Model_Metrices_Table b ,Variable_importance c
WHERE a.ID=c.Model_id and a.ID=b.ID;
```

ID	Model_id	RMSE
GBM_31	GBM_grid_1_AutoML_20190419_013027_model_7	0.13394528800
XBG_1	XGBoost_3_AutoML_20190419_003540	0.13543318300
XBG_13	XGBoost_3_AutoML_20190419_005900	0.13504922400

7.2 Find top 10 deeplearning model to find their evaluation performance.

```
SELECT a.ID,a.Model_id,b.RMSE
FROM Model_information a, Model_Metrices_Table b
WHERE Model_id like 'Deep%' and a.ID= b.ID
ORDER BY RMSE
limit 10;
```

ID	Model_id	RMSE
DL_1	DeepLearning_grid_1_AutoML_20190419_003540_model_1	0.16199104000
DL_12	DeepLearning_grid_1_AutoML_20190419_003540_model_1	0.16199104000
DL_4	DeepLearning_grid_1_AutoML_20190419_003540_model_1	0.16199104000
DL_13	DeepLearning_grid_1_AutoML_20190419_005900_model_1	0.16713995700
DL_5	DeepLearning_grid_1_AutoML_20190419_005900_model_1	0.16713995700
DL_14	DeepLearning_grid_1_AutoML_20190419_005900_model_4	0.16717610000
DL_6	DeepLearning_grid_1_AutoML_20190419_005900_model_4	0.16717610000
DL_15	DeepLearning_grid_1_AutoML_20190419_013027_model_3	0.16857485000
DL_2	DeepLearning_grid_1_AutoML_20190419_003540_model_2	0.17142232000
DL_7	DeepLearning_grid_1_AutoML_20190419_003540_model_2	0.17142232000

7.3 Find one type of models to calculate its average RMSE value.

```
SELECT AVG(b.RMSE) as 'average performance'
FROM Model_information a, Model_Metrices_Table b
WHERE Model_id like 'GBM%'and a.ID= b.ID;
```

average performance
0.166881896823529

7.4 Find one model's one parameters.

```
SELECT a.Model_id,b.max_depth_actual
FROM Model_information a, XGBoost_Model b
WHERE a.ID= b.Model_id
ORDER BY max_depth_actual;
```

Model_id	max_depth_actual
XGBoost_grid_1_AutoML_20190419_013027_model_8	5
XGBoost_grid_1_AutoML_20190419_005900_model_3	5
XGBoost_grid_1_AutoML_20190419_003540_model_7	5
XGBoost_grid_1_AutoML_20190419_003540_model_9	5
XGBoost_grid_1_AutoML_20190419_005900_model_5	5
XGBoost_grid_1_AutoML_20190419_003540_model_3	5
XGBoost_grid_1_AutoML_20190419_005900_model_6	5
XGBoost_grid_1_AutoML_20190419_005900_model_2	5
XGBoost_grid_1_AutoML_20190419_013027_model_13	5
XGBoost_grid_1_AutoML_20190419_003540_model_3	5
XGBoost_grid_1_AutoML_20190419_003540_model_9	5
XGBoost_grid_1_AutoML_20190419_005900_model_3	5
XGBoost_grid_1_AutoML_20190419_003540_model_7	5
XGBoost_grid_1_AutoML_20190419_003540_model_9	5
XGBoost_grid_1_AutoML_20190419_013027_model_4	5

7.5 Find one model's RMSE value and its hyperparameters.

```
SELECT mm.ID, mm.Model_id, mm.Run_time,
me.RMSE,
dl. activation_actual, dl.rho_actual,epochs_actual
FROM Model_information mm
left join Deep_learning_model dl
ON mm.ID = dl.Model_id
JOIN Model_Metrices_Table me
ON  dl.Model_id = me.ID
WHERE mm.ID LIKE "DL%"
ORDER BY  mm.Run_time;
```

ID	Model_id	Run_time	RMSE
DL_1	DeepLearning_grid_1_AutoML_20190419_003540_model_1	500	0.161991
DL_2	DeepLearning_grid_1_AutoML_20190419_003540_model_2	500	0.171422
DL_3	DeepLearning_1_AutoML_20190419_003540	500	0.185573
DL_5	DeepLearning_grid_1_AutoML_20190419_005900_model_1	1000	0.167139
DL_6	DeepLearning_grid_1_AutoML_20190419_005900_model_4	1000	0.167176
DL_10	DeepLearning_grid_1_AutoML_20190419_005900_model_3	1000	0.189775
DL_7	DeepLearning_grid_1_AutoML_20190419_003540_model_2	1000	0.171422
DL_11	DeepLearning_grid_1_AutoML_20190419_005900_model_2	1000	0.222808
DL_8	DeepLearning_1_AutoML_20190419_003540	1000	0.185573
DL_4	DeepLearning_grid_1_AutoML_20190419_003540_model_1	1000	0.161991
DL_9	DeepLearning_1_AutoML_20190419_005900	1000	0.185727
DL_13	DeepLearning_grid_1_AutoML_20190419_005900_model_1	1500	0.167139
DL_18	DeepLearning_1_AutoML_20190419_003540	1500	0.185573
DL_22	DeepLearning_grid_1_AutoML_20190419_013027_model_1	1500	0.200932
DL_14	DeepLearning_grid_1_AutoML_20190419_005900_model_4	1500	0.167176

7.6 Find the model and its RMSE value which is less than one model's average RMSE value.

```
SELECT a.Model_id,b.RMSE
FROM Model_information a, Model_Metrices_Table b
WHERE a.ID=b.ID and b.RMSE<(
SELECT AVG(b.RMSE)
FROM Model_information a, Model_Metrices_Table b
WHERE Model_id like 'G%'and a.ID= b.ID)
ORDER BY b.RMSE
LIMIT 10;
```

Model_id	RMSE
GBM_grid_1_AutoML_20190419_013027_model_7	0.13394528800
XGBoost_3_AutoML_20190419_005900	0.13504922400
XGBoost_3_AutoML_20190419_005900	0.13504922400
XGBoost_grid_1_AutoML_20190419_005900_model_12	0.13516583500
XGBoost_grid_1_AutoML_20190419_005900_model_12	0.13516583500
XGBoost_grid_1_AutoML_20190419_005900_model_14	0.13519683100
XGBoost_grid_1_AutoML_20190419_005900_model_14	0.13519683100
XGBoost_grid_1_AutoML_20190419_013027_model_13	0.13538536700
XGBoost_grid_1_AutoML_20190419_005900_model_4	0.13542368600
XGBoost_grid_1_AutoML_20190419_005900_model_4	0.13542368600

7.7 When the run time is 1000, find the most important predictors which percentage is more than 0.2.

```
SELECT b.Variable
FROM Variable_importance b
where b.Percentage>0.2 and b.Model_id=(
SELECT distinct c.ID
FROM Model_information c,Variable_importance d
WHERE c.ID=d.Model_id and c.Run_time=1000);
```

Variable
OverallQual
GrLivArea

7.8 Find the best model's top five important variables and their percentage.

```
select Model_id, Variable,Percentage
from Variable_importance
where (
    select count(*) from Variable_importance as f
    where f.Model_id = Variable_importance.Model_id
   AND f.Percentage > Variable_importance.Percentage)<=4
   ORDER BY Model_id;
```

Model_id	Variable	Percentage
GBM_31	OverallQual	0.2094596520
GBM_31	GrLivArea	0.2049968480
GBM_31	TotalBsmtSF	0.1115365460
GBM_31	YearRemodAdd	0.0471078040
GBM_31	YearBuilt	0.0432619980
XBG_1	OverallQual	0.2311314680
XBG_1	GrLivArea	0.1996154960
XBG_1	TotalBsmtSF	0.0904063690
XBG_1	GarageCars	0.0467541870
XBG_1	YearRemodAdd	0.0466488300
XBG_13	OverallQual	0.2094596520
XBG_13	GrLivArea	0.2049968480
XBG_13	TotalBsmtSF	0.1115365460
XBG_13	YearRemodAdd	0.0471078040
XBG_13	YearBuilt	0.0432619980

7.9 How many models has run for 1500.

```
SELECT COUNT(*) as " the number of 1500 runtime model"
FROM Model_information mm
WHERE mm.Run_time = 1500
order BY mm.Run_time;
```

the number of 1500 runtime model
102

7.10 Calculate two model's average RMSE to find which model is better.

```
SELECT DISTINCT (
SELECT AVG(a.RMSE) FROM Model_Metrices_Table a WHERE a.ID like "DL%"
) - (
SELECT AVG(a.RMSE) FROM Model_Metrices_Table a WHERE a.ID like "XBG%"
) AS difference_between_two_model
FROM Model_Metrices_Table;
```

difference_between_two_model
-1.279394658379710

8. VIEWS:

8.1 Get top model performance

```
CREATE VIEW TOP_MODEL AS
SELECT ID,RMSE
FROM Model_Metrices_Table
ORDER BY RMSE DESC
LIMIT 5;
```

ID	RMSE
XBG_90	10.00280743000
XBG_41	10.00280743000
XBG_89	7.95016036400
XBG_88	7.53286680900
XBG_40	7.53286680900

8.2 Get model information on run time and performance

```
CREATE VIEW Model_RMSE AS
SELECT Model_id,RMSE,Run_time
FROM Model_information a,Model_Metrices_Table b
WHERE a.ID=b.ID;
```

Model_id	variable	Percentage	Run_time
XGBoost_3_AutoML_20190419_005900	OverallQual	0.2094596520	1000
XGBoost_3_AutoML_20190419_005900	GrLivArea	0.2049968480	1000
XGBoost_3_AutoML_20190419_005900	TotalBsmtSF	0.1115365460	1000
XGBoost_3_AutoML_20190419_005900	YearRemodAdd	0.0471078040	1000
XGBoost_3_AutoML_20190419_005900	YearBuilt	0.0432619980	1000
XGBoost_3_AutoML_20190419_005900	GarageCars	0.0396015120	1000
XGBoost_3_AutoML_20190419_005900	BsmtFinSF1	0.0361869510	1000
XGBoost_3_AutoML_20190419_005900	Fireplaces	0.0351197000	1000
XGBoost_3_AutoML_20190419_005900	LotArea	0.0328989840	1000
XGBoost_3_AutoML_20190419_005900	GarageArea	0.0317572180	1000
XGBoost_3_AutoML_20190419_005900	1stFlrSF	0.0218631000	1000
XGBoost_3_AutoML_20190419_005900	GarageYrBlt	0.0201163300	1000
XGBoost_3_AutoML_20190419_005900	LotFrontage	0.0143102570	1000
XGBoost_3_AutoML_20190419_005900	CentralAir_Y	0.0135843050	1000
XGBoost_3_AutoML_20190419_005900	FireplaceQu_N…	0.0098935510	1000
XGBoost_3_AutoML_20190419_005900	GarageType_A…	0.0083199130	1000
XGBoost_3_AutoML_20190419_005900	ExterQual_TA	0.0082624670	1000

8.3 Get everydl model hyper in the db

```
CREATE VIEW DL_Hyper AS
SELECT a.Model_id,epochs_actual,RMSE
FROM Model_information a,Model_Metrices_Table b,Deep_learning_model c
WHERE a.ID=b.ID AND b.ID=c.Model_id AND c.Model_id=a.ID;
```

Model_id	epochs_actual	RMSE
DeepLearning_grid_1_AutoML_20190419_003...	8.000000000	0.16199104000
DeepLearning_grid_1_AutoML_20190419_005...	126.400000000	0.18977505100
DeepLearning_grid_1_AutoML_20190419_005...	8.000000000	0.22280865600
DeepLearning_grid_1_AutoML_20190419_003...	8.000000000	0.16199104000
DeepLearning_grid_1_AutoML_20190419_005...	307.200000000	0.16713995700
DeepLearning_grid_1_AutoML_20190419_005...	8.063013699	0.16717610000
DeepLearning_grid_1_AutoML_20190419_013...	238.400000000	0.16857485000
DeepLearning_grid_1_AutoML_20190419_003...	5.117123288	0.17142232000
DeepLearning_1_AutoML_20190419_013027	10.377054790	0.17714048500
DeepLearning_1_AutoML_20190419_003540	10.393664380	0.18557315900
DeepLearning_1_AutoML_20190419_005900	10.427910960	0.18572720300
DeepLearning_grid_1_AutoML_20190419_003...	5.117123288	0.17142232000
DeepLearning_grid_1_AutoML_20190419_005...	126.400000000	0.18977505100
DeepLearning_grid_1_AutoML_20190419_013...	11.192979450	0.20022714800
DeepLearning_grid_1_AutoML_20190419_013...	291.200000000	0.20093231200
DeepLearning_grid_1_AutoML_20190419_005...	8.000000000	0.22280865600
DeepLearning_1_AutoML_20190419_003540	10.393664380	0.18557315900

8.4 Get run time and variable importance of model.

```
CREATE VIEW Model_Variable_importance AS
SELECT a.Model_id,variable,Percentage,b.Run_time
FROM Model_information a,Variable_importance b
WHERE a.ID=b.Model_id;
```

Model_id	RMSE	Run_time
DeepLearning_grid_1_AutoML_20190419_003...	0.16199104000	500
DeepLearning_grid_1_AutoML_20190419_005...	0.18977505100	1000
DeepLearning_grid_1_AutoML_20190419_005...	0.22280865600	1000
DeepLearning_grid_1_AutoML_20190419_003...	0.16199104000	1500
DeepLearning_grid_1_AutoML_20190419_005...	0.16713995700	1500
DeepLearning_grid_1_AutoML_20190419_005...	0.16717610000	1500
DeepLearning_grid_1_AutoML_20190419_013...	0.16857485000	1500
DeepLearning_grid_1_AutoML_20190419_003...	0.17142232000	1500
DeepLearning_1_AutoML_20190419_013027	0.17714048500	1500
DeepLearning_1_AutoML_20190419_003540	0.18557315900	1500
DeepLearning_1_AutoML_20190419_005900	0.18572720300	1500
DeepLearning_grid_1_AutoML_20190419_003...	0.17142232000	500
DeepLearning_grid_1_AutoML_20190419_005...	0.18977505100	1500

9.fuction:

9.1 Get moedl id

```
DELIMITER $$
CREATE FUNCTION getmodelid ( v_id VARCHAR(30)) RETURNS VARCHAR ( 255 ) BEGIN
    DECLARE
        modelid VARCHAR ( 255 );
    SELECT
        model_id INTO modelid
    FROM
        model_information
    WHERE
        ID = v_id;
    RETURN modelid;

END $$
```

9.2 Get performance of deep learning

```
DELIMITER $$
CREATE FUNCTION dlperformance() RETURNS VARCHAR(25) BEGIN
  DECLARE
        a VARCHAR(25);
    SELECT
        AVG(RMSE) INTO a
    FROM
        model_metrices_table
    WHERE
        ID LIKE 'DL%';
    RETURN a;

END $$
```

9.3 Get number of model smaller than RMSE

```
DELIMITER $$
CREATE FUNCTION NUMRMSE(num DECIMAL(5)) RETURNS INT BEGIN
  DECLARE
        a INT;
    SELECT
        COUNT(*) INTO a
    FROM
        model_metrices_table
    WHERE
        RMSE < num;
    RETURN a;

END $$
```

9.4 Get number of model in runtime

```
DELIMITER $$
CREATE FUNCTION runtimemodel(num INT) RETURNS INT BEGIN
  DECLARE
        a INT;
    SELECT
        COUNT(*) INTO a
    FROM
        model_information
    WHERE
        Run_time = num;
    RETURN a;

END $$
```

10. analysis

From the database, we compare different types of algorithms. Among all these algorithms , XGBOOST perform best on predict the house price. When we handle the regression mission , the best standard for us to evaluate is the RMSE.

11. conclusion

During the project, we create the database to store the data and the model generate from the H2O platform. 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.

12.citataion and reference

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

13.MIT License

Copyright (c) 2019 Quanhan Sun, Guoyan Li

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.