# Easier Data Preprocessing in PostgreSQL

資料科學碩一 張烱郁
資工所碩一　　曾尹均
資工所碩一　　簡丞珮
經濟所碩二　　林祐辰

# Outline

- Introduction to our project
- Implemented functions
  - Missing values filling
  - Feature selection for regression and classification tasks
  - Outlier detection
- Applicaton DEMO--House price prediction

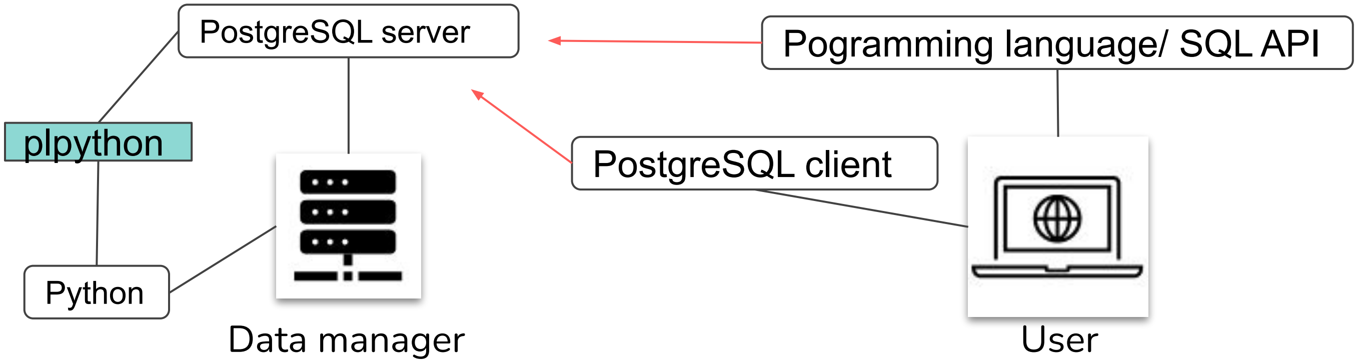
# Introduction

# Our goal

We want to solve the problem for those who want to:

1. easily get a **complete table**
2. quickly find **important variables**
3. find and exclude **abnormal instances**
4. **reduce I/O** of data when accessing a remote server
5. **get rid of** dealing with **categorical and numerical** variables

# Our implemtation framework



# Missing values filling

# Missing values

| | crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | b | lstat | price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **20** | NaN | 55.0 | 3.78 | None | 0.484 | 6.696 | NaN | 5.7321 | 5.0 | 370.0 | 17.6 | 396.90 | 7.18 | 23.9 |
| **21** | 37.66190 | 0.0 | 18.10 | False | 0.679 | 6.202 | 78.7 | 1.8629 | 24.0 | 666.0 | 20.2 | 18.82 | 14.52 | 10.9 |
| **22** | NaN | 0.0 | 13.92 | False | NaN | 6.009 | NaN | 5.5027 | 4.0 | 289.0 | 16.0 | 396.90 | 10.40 | 21.7 |
| **23** | 0.22438 | 0.0 | 9.69 | False | 0.585 | 6.027 | 79.7 | 2.4982 | 6.0 | 391.0 | 19.2 | 396.90 | 14.33 | 16.8 |
| **24** | 5.66998 | 0.0 | 18.10 | True | 0.631 | 6.683 | 96.8 | 1.3567 | 24.0 | 666.0 | 20.2 | 375.33 | 3.73 | 50.0 |
| **25** | 0.04819 | 80.0 | 3.64 | False | 0.392 | 6.108 | 32.0 | 9.2203 | 1.0 | 315.0 | 16.4 | 392.89 | 6.57 | 21.9 |
| **26** | 1.42502 | NaN | 19.58 | False | 0.871 | NaN | 100.0 | 1.7659 | 5.0 | 403.0 | NaN | 364.31 | 7.39 | 23.3 |
| **27** | 0.06888 | 0.0 | 2.46 | False | 0.488 | 6.144 | 62.2 | 2.5979 | 3.0 | 193.0 | 17.8 | 396.90 | 9.45 | 36.2 |
| **28** | 0.06162 | 0.0 | 4.39 | False | 0.442 | 5.898 | 52.3 | 8.0136 | 3.0 | 352.0 | 18.8 | 364.61 | 12.67 | 17.2 |
| **29** | 11.08740 | 0.0 | 18.10 | False | 0.718 | 6.411 | 100.0 | 1.8589 | 24.0 | 666.0 | 20.2 | 318.75 | 15.02 | 16.7 |

categorical + numerical ? Don't worry!

# Alogorithm

Use information from other attributes to predict the missing values.

| | crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | b | lstat | price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **20** | NaN | 55.0 | 3.78 | None | 0.484 | 6.696 | NaN | 5.7321 | 5.0 | 370.0 | 17.6 | 396.90 | 7.18 | 23.9 |
| **21** | 37.66190 | 0.0 | 18.10 | False | 0.679 | 6.202 | 78.7 | 1.8629 | 24.0 | 666.0 | 20.2 | 18.82 | 14.52 | 10.9 |
| **22** | NaN | 0.0 | 13.92 | False | NaN | 6.009 | NaN | 5.5027 | 4.0 | 289.0 | 16.0 | 396.90 | 10.40 | 21.7 |
| **23** | 0.22438 | 0.0 | 9.69 | False | 0.585 | 6.027 | 79.7 | 2.4982 | 6.0 | 391.0 | 19.2 | 396.90 | 14.33 | 16.8 |
| **24** | 5.66998 | 0.0 | 18.10 | True | 0.631 | 6.683 | 96.8 | 1.3567 | 24.0 | 666.0 | 20.2 | 375.33 | 3.73 | 50.0 |
| **25** | 0.04819 | 80.0 | 3.64 | False | 0.392 | 6.108 | 32.0 | 9.2203 | 1.0 | 315.0 | 16.4 | 392.89 | 6.57 | 21.9 |
| **26** | 1.42502 | NaN | 19.58 | False | 0.871 | NaN | 100.0 | 1.7659 | 5.0 | 403.0 | NaN | 364.31 | 7.39 | 23.3 |
| **27** | 0.06888 | 0.0 | 2.46 | False | 0.488 | 6.144 | 62.2 | 2.5979 | 3.0 | 193.0 | 17.8 | 396.90 | 9.45 | 36.2 |
| **28** | 0.06162 | 0.0 | 4.39 | False | 0.442 | 5.898 | 52.3 | 8.0136 | 3.0 | 352.0 | 18.8 | 364.61 | 12.67 | 17.2 |
| **29** | 11.08740 | 0.0 | 18.10 | False | 0.718 | 6.411 | 100.0 | 1.8589 | 24.0 | 666.0 | 20.2 | 318.75 | 15.02 | 16.7 |

underlying model:
histogram-based gradient boosting

Use a regression model to predict

Use a classification model to predict

# Usage

| | crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | b | lstat | price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **20** | NaN | 55.0 | 3.78 | None | 0.484 | 6.696 | NaN | 5.7321 | 5.0 | 370.0 | 17.6 | 396.90 | 7.18 | 23.9 |
| **21** | 37.66190 | 0.0 | 18.10 | False | 0.679 | 6.202 | 78.7 | 1.8629 | 24.0 | 666.0 | 20.2 | 18.82 | 14.52 | 10.9 |
| **22** | NaN | 0.0 | 13.92 | False | NaN | 6.009 | NaN | 5.5027 | 4.0 | 289.0 | 16.0 | 396.90 | 10.40 | 21.7 |
| **23** | 0.22438 | 0.0 | 9.69 | False | 0.585 | 6.027 | 79.7 | 2.4982 | 6.0 | 391.0 | 19.2 | 396.90 | 14.33 | 16.8 |
| **24** | 5.66998 | 0.0 | 18.10 | True | 0.631 | 6.683 | 96.8 | 1.3567 | 24.0 | 666.0 | 20.2 | 375.33 | 3.73 | 50.0 |
| **25** | 0.04819 | 80.0 | 3.64 | False | 0.392 | 6.108 | 32.0 | 9.2203 | 1.0 | 315.0 | 16.4 | 392.89 | 6.57 | 21.9 |
| **26** | 1.42502 | NaN | 19.58 | False | 0.871 | NaN | 100.0 | 1.7659 | 5.0 | 403.0 | NaN | 364.31 | 7.39 | 23.3 |
| **27** | 0.06888 | 0.0 | 2.46 | False | 0.488 | 6.144 | 62.2 | 2.5979 | 3.0 | 193.0 | 17.8 | 396.90 | 9.45 | 36.2 |
| **28** | 0.06162 | 0.0 | 4.39 | False | 0.442 | 5.898 | 52.3 | 8.0136 | 3.0 | 352.0 | 18.8 | 364.61 | 12.67 | 17.2 |
| **29** | 11.08740 | 0.0 | 18.10 | False | 0.718 | 6.411 | 100.0 | 1.8589 | 24.0 | 666.0 | 20.2 | 318.75 | 15.02 | 16.7 |

| | crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | b | lstat | price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **20** | 0.069066 | 55.000000 | 3.78 | False | 0.484000 | 6.696000 | 54.031289 | 5.7321 | 5.0 | 370.0 | 17.600000 | 396.90 | 7.18 | 23.9 |
| **21** | 37.661900 | 0.000000 | 18.10 | False | 0.679000 | 6.202000 | 78.700000 | 1.8629 | 24.0 | 666.0 | 20.200000 | 18.82 | 14.52 | 10.9 |
| **22** | -0.363259 | 0.000000 | 13.92 | False | 0.484053 | 6.009000 | 53.844987 | 5.5027 | 4.0 | 289.0 | 16.000000 | 396.90 | 10.40 | 21.7 |
| **23** | 0.224380 | 0.000000 | 9.69 | False | 0.585000 | 6.027000 | 79.700000 | 2.4982 | 6.0 | 391.0 | 19.200000 | 396.90 | 14.33 | 16.8 |
| **24** | 5.669980 | 0.000000 | 18.10 | True | 0.631000 | 6.683000 | 96.800000 | 1.3567 | 24.0 | 666.0 | 20.200000 | 375.33 | 3.73 | 50.0 |
| **25** | 0.048190 | 80.000000 | 3.64 | False | 0.392000 | 6.108000 | 32.000000 | 9.2203 | 1.0 | 315.0 | 16.400000 | 392.89 | 6.57 | 21.9 |
| **26** | 1.425020 | -1.150642 | 19.58 | False | 0.871000 | 6.256387 | 100.000000 | 1.7659 | 5.0 | 403.0 | 14.566458 | 364.31 | 7.39 | 23.3 |
| **27** | 0.068880 | 0.000000 | 2.46 | False | 0.488000 | 6.144000 | 62.200000 | 2.5979 | 3.0 | 193.0 | 17.800000 | 396.90 | 9.45 | 36.2 |
| **28** | 0.061620 | 0.000000 | 4.39 | False | 0.442000 | 5.898000 | 52.300000 | 8.0136 | 3.0 | 352.0 | 18.800000 | 364.61 | 12.67 | 17.2 |
| **29** | 11.087400 | 0.000000 | 18.10 | False | 0.718000 | 6.411000 | 100.000000 | 1.8589 | 24.0 | 666.0 | 20.200000 | 318.75 | 15.02 | 16.7 |

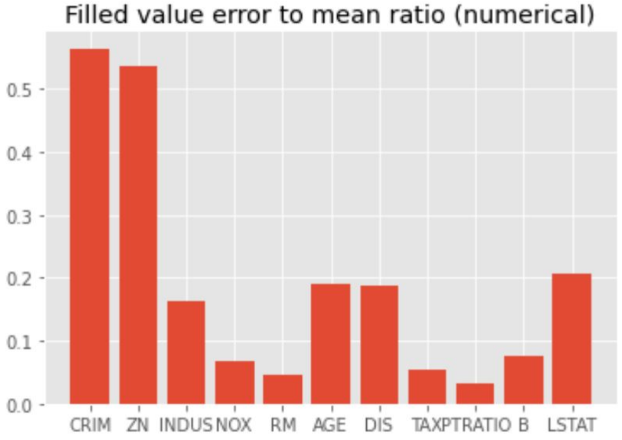
```sql
SELECT fill_missing('boston_miss', 'boston_miss_filled', 'price');
```

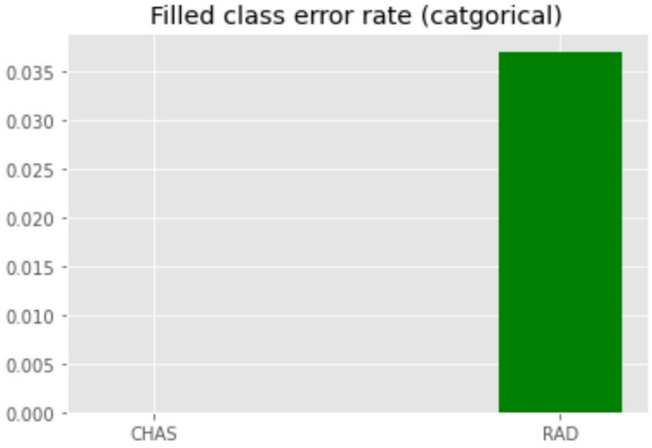
input table

output table

target column (optional)

# Performance



# Feature Selection

# Methods

- Filter based
  - SelectKBest
  - SelectPercentile

- Wrapper-based
  - RFE

- Embedded
  - SelectFromModel

# SelectKbest

```
CREATE OR REPLACE FUNCTION SelectKBest (
                                table_name varchar(63),
                                save_as varchar(63),
                                target_name varchar(128),
                                score_func varchar(22) default 'f_classif',
                                k int default 10)
RETURNS text[]
```

# SelectPercentile

```
CREATE OR REPLACE FUNCTION SelectPercentile (
                                table_name varchar(63),
                                save_as varchar(63),
                                target_name varchar(128),
                                cat_names varchar(128)[] default null,
                                percentile int default 10)
RETURNS text[]
```

# RFE

```
CREATE OR REPLACE FUNCTION RFE (
                            table_name varchar(63),
                            save_as varchar(63),
                            target_name varchar(128),
                            estimator_name text,
                            n_features_to_select real default null,
                            step real default 1,
                            verbose_ int default 0,
                            importance_getter text default 'auto')
RETURNS text[]
```

# SelectFromModel

```sql
CREATE OR REPLACE FUNCTION SelectFromModel (
                                table_name varchar(63),
                                save_as varchar(63),
                                target_name varchar(63),
                                estimator_name varchar(63) default 'LASSO',
                                threshold varchar(63) default null,
                                max_features INT default null,
                                importance_getter text default 'auto')
RETURNS text[]
```

# Result

```
jcp=# SELECT RFE('boston', 'selected_view', 'price', 'GradientBoostingRegressor');
                rfe
-----------------------------------
 {crim,nox,rm,dis,ptratio,lstat}
(1 row)
```

# Outlier Detection

```sql
CREATE OR REPLACE FUNCTION OutlierDetection (table_name varchar(63),
                                             method varchar(63) default 'OCSVM',
                                             target varchar(63) default 'NULL')
RETURNS SETOF INT
AS $$
```

# Parameters

Table_name: Table name

Method:
OneClassSVM/ Isolation Forest/
Minimum Covariance Determinant/
DBSCAN

Target: column name of target variable, would not be used used to detect outliers

```sql
select OutlierDetection('Boston', 'DBSCAN', 'PRICE');
```

| | outlierdetection integer |
|---|---|
| 1 | 17 |
| 2 | 65 |
| 3 | 105 |
| 4 | 115 |
| 5 | 117 |
| 6 | 118 |
| 7 | 125 |
| 8 | 126 |
| 9 | 177 |
| 10 | 246 |

```sql
--add unique row_idx
alter table Boston add unique_id serial;

select * from Boston
where unique_id not in ((select OutlierDetection('Boston', 'OCSVM')));
```

# Demo

# Video Download Link

https://drive.google.com/file/d/16cj54VMcMqKj1tVexFC0ke-E2LDj_trE/view?usp=sharing