Module 4: Data Preparation and Modeling

Day on the life of a Data Scientist Workshop

Objectives

After completing this module, you will be able to do the following:

- Write queries in SQL, Python, and R to transform data and use it to build a model
- Discuss Data Science Process as applicable

Topics

- Introduction
- House Pricing Model
    - Data Science Process
        - SQL
        - Python
        - R
- Summary and Review

Current Topic – Introduction

- **Introduction**
- House Pricing Model
  - Data Science Process
    - SQL
    - Python
    - R
- Summary and Review

Introduction

- In this module, we will be using various analytic functions in **SQL**, **Python**, and **R** related to computer hardware data
- We will first prepare our source data by removing outliers and scaling certain variables. We will then build a Model to predict the prices for Houses in Colombia. Lastly, we will quantify the accuracy of the Model.

Current Topic – House Pricing Model: Data Science Process

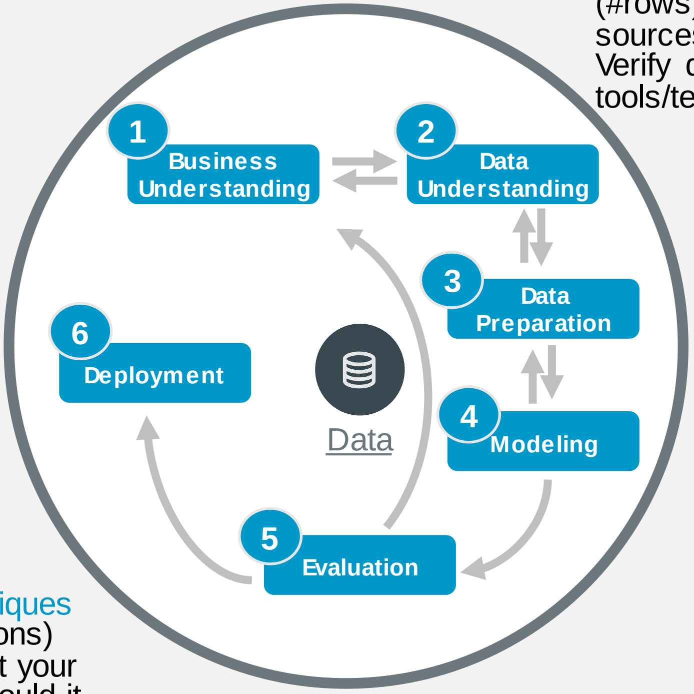
- Introduction
- **House Pricing Model**
  - **Data Science Process**
    - SQL
    - Python
    - R
- Summary and Review

Data Science Process (CRISP DM) – Six Steps

1. You must be clear on the business goal of the analytic request, the success criteria and Data Science goal(s)

6. Operationalize and revisit over time. Should it be operationalized? Once you have followed your companies process for operationalizing, you need to consider these monitoring and maintenance questions: Has data changed? Has new data become available? Is the model still predicting accurately? Etc.

5. Compare models / analysis techniques (e.g. using Vantage statistical functions) and select best results. Does it meet your business goals/success criteria? Should it be visualized and/or operationalized?



CRISP-DM

2. You must know the underlying data. Size (#rows)? What's in the data (Columns)? Data sources? Schema type - structured or unstructured? Verify data quality. Use Vantage functions and other tools/techniques to examine the data

3. Does the data need preparing? Yes/No? Remove outliers? Missing data? Scale? Transform? Organize by geolocation? Perform data preparation using Vantage data preparation functions

4. Which predictive/analytic functions? Which arguments? Based on function, does data need further cleaning/preparing? Execute Vantage ML functions to create models/ analyses

- Build supervised/predictive model(s) on training set, validate on test set, and use Vantage statistical functions to assess accuracy of results
- Perform unsupervised/descriptive analytics on full data set and assess reasonableness of results visually
- Make sure to experiment. Use visualization to assist in model/analysis assessment. Keep track of peripheral findings

Step 1. Business Understanding
Data Science Process for the House Pricing Model

Step	Description	Activities	Comments
1	**Business Understanding**	Business goal? Success criteria? DS goal?	We are a real estate company, and we would like to build a model that accurately predicts the prices for houses in Colombia. In this case study, we will use SQL, Python, and R
2	**Data Understanding**	What does the data show? Where is it located? Is it complete/correct?	Data resides in **Precio_Casas_Col**, which contains detailed information of various ads in a home sales website. We will illustrate how to remove outliers and perform scaling
3	**Data Preparation**	Outliers? Scale? Organize by geolocation?	We will remove outliers and perform scaling on chosen variables
4	**Modeling/ Analysis**	Which functions? Which arguments? Experiment! Assess and compare models/analyses	Use available algorithms in VAL
5	**Evaluation**	Are your business goals and criteria being met? Visualize? Pick best performers	Analyze the results and how accurately our model predicts published prices
6	**Deployment**	Operationalize and revisit over time.	Plan to operationalize the analytic results varies by customer and is not covered in this course.

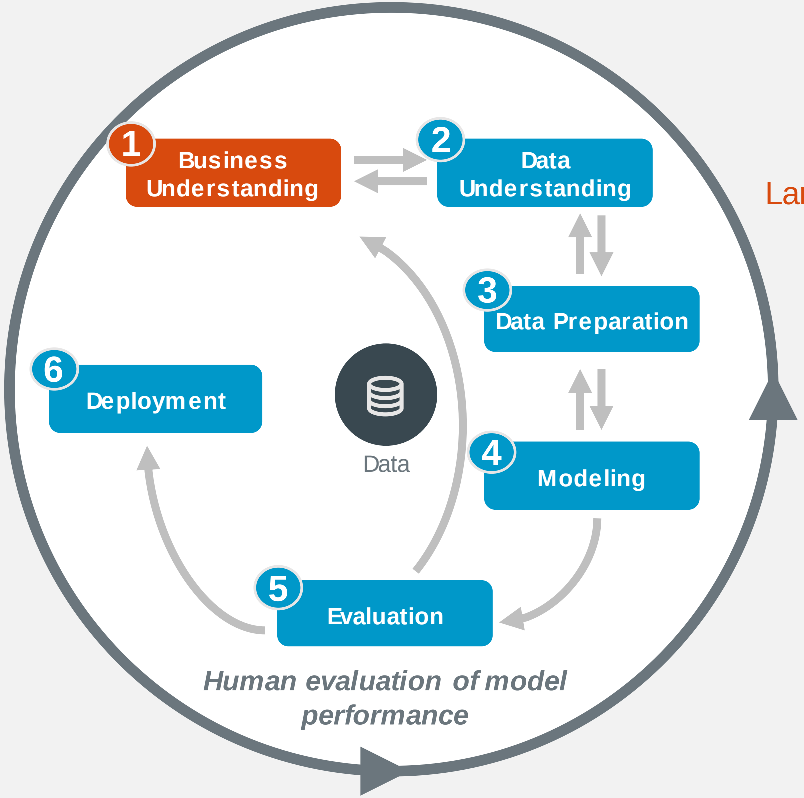
Current Topic – House Pricing Model with Data Science Process (SQL)

teradata.

- Introduction
- **House Pricing Model**
  - **Data Science Process**
    - **SQL**
    - Python
    - R
- Summary and Review

Step 1. Business Understanding
Map Out Data Science Process – Tools/Functions

Business Objective:
Build a Model that accurately predicts published House Prices.
Prepare data first



CRISP-DM

Language: SQL in Teradata Studio

Data Preparation Functions: dataexplorer, vartran

Modeling Functions: linear

Visualization apps: N/A

Step 2. Data Understanding
Lab 01: View the Raw Data

Business Objective: Build a **Linear Regression** Model to predict House Prices. Prepare data by first modifying Outlier values and Transforming the data

```
SELECT * FROM Precio_Casas_Col;
```

	banos	ba...	co...	cuar...	cu...	dep...	dep...	estrato	est...	ga...	gar...	gim...	habitaciones	ha...	instala...	jac...	jar...	latitud	longitud	nu...	pa...	pis...	pla...	porter...	remo...	salonc...	sau...	terraza	tiempodeconstr...	tipodegaraj	valor
1	2	null	Si	null	null	1	1	4	null	Si	1	Si	1	null	Natural	null	null	4.63965	-74.06217	1	null	null	null	24hrs	No	null	null	null	Entre 0 y 5 años	Independiente	270675000.00000
2	2	null	null	null	null	null	null	4	null	Si	2	Si	2	null	null	null	null	4.72317	-74.03037	1	null	null	null	null	No	null	null	Balcón	Entre 0 y 5 años	Independiente	430000000.00000
3	1	null	Si	null	null	null	null	5	null	Si	1	null	1	null	null	null	null	4.71076	-74.04765	null	null	null	null	null	No	null	null	null	Entre 0 y 5 años	Propio	300000000.00000
4	2	null	null	null	null	1	1	4	null	Si	2	null	1	null	Natural	null	null	4.64847	-74.05513	2	null	null	null	24hrs	No	Si	null	Balcón	Entre 0 y 5 años	Servidumbre	430000000.00000
5	2	null	null	null	null	1	1	5	null	Si	2	null	2	null	Natural	null	null	4.69499	-74.06491	null	null	null	null	null	No	null	null	Ninguno	Entre 0 y 5 años	Servidumbre	430000000.00000
6	2	null	null	null	null	1	1	5	null	Si	1	Si	1	null	Natural	null	null	4.71950	-74.05031	1	null	null	null	24hrs	No	Si	null	Balcón	Entre 0 y 5 años	Independiente	285000000.00000
7	3	null	Si	null	Si	1	1	4	null	Si	2	Si	3	Si	Natural	null	null	4.72420	-74.03983	1	null	null	null	24hrs	No	Si	null	null	Entre 0 y 5 años	Propio	380000000.00000
8	3	null	null	null	null	null	null	null	null	null	null	null	3	null	null	null	null	4.69297	-74.05210	null	null	null	null	null	No	null	null	null	Entre 0 y 5 años	null	1049334000.0000
9	2	null	null	null	null	null	null	4	null	Si	2	null	2	Si	Natural	null	null	4.71787	-74.06957	null	null	null	null	24hrs	No	Si	null	null	Entre 0 y 5 años	null	390000000.00000
10	4	null	Si	null	Si	1	1	6	Si	Si	3	null	3	Si	Natural	null	null	4.71267	-74.02573	1	null	null	null	24hrs	No	null	null	null	Más de 20 años	Propio	1590000000.0000

- Describe and explore data
- Verify data adequacy/quality

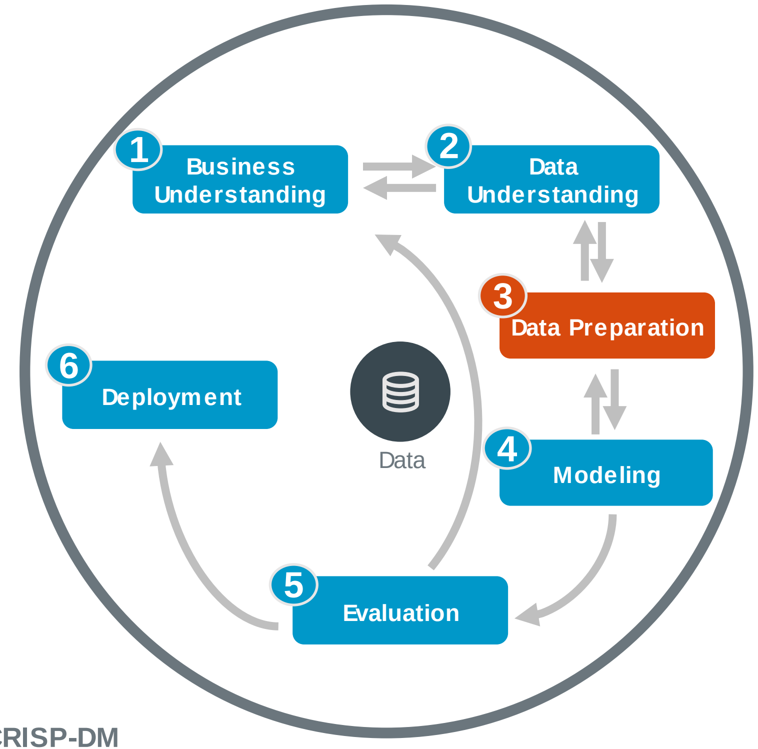
Table houses various characteristics related to houses. The id is the unique identifier. Our dependent variable is valor

Yes, data is adequate for analysis task
Complete? – Yes, covers all cases required
Correct? – Yes, no missing values or errors

Step 3. Data Preparation

a) Does data require Cleaning? Does data need to be Scaled? Do Outliers need to be removed?

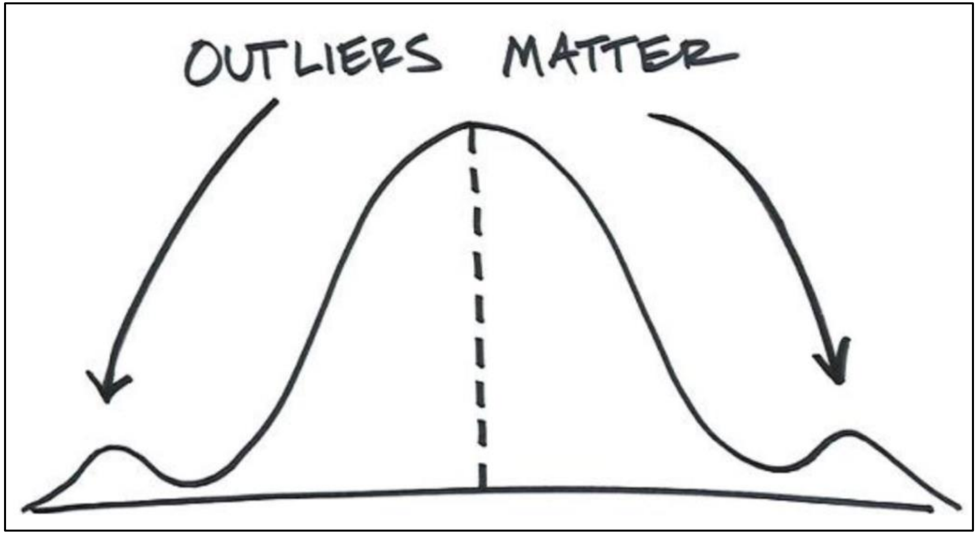
Yes. We will remove outliers and scale desired variables



CRISP-DM

Step 3. Data Preparation
Identify Outliers

- Detecting and handling outliers becomes critical for certain statistical measures
- Analysis of data may not be meaningful if outliers are in the data



The **DataExplorer, Values, Statistics** options might be useful to detect abnormal and missing values; e.g., it could be useful for identifying the very best of your customers, it could be used to detect fraudulent activity, etc.

Step 3. Data Preparation
Values Syntax


```
call td_analyze(‘values’,
'database = val_source;
Tablename = customer;
Columns = all;');
```

Step 3. Data Preparation
Lab 02a: Exploring the Dataset

teradata.


```
-- Exploring the Values in the dataset
call TRNG_XSP.td_analyze (
'values',
'database = ADLSLSAMER_MS_AZ;
tablename = Precio_Casas_Col;
columns = all;');
```

- We can see the count of nulls, unique values, positive, negative, zeros, etc.

	xdb	xtbl	xcol	xtype	xcnt	xnull	xunique	xblank	xzero	xpos	xneg
1	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonasverdes	VARCHAR(2) CHARACTER SET LATIN	145552.00000	95643.00000	1.00000	0.00000	null	null	null
2	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonaninos	VARCHAR(2) CHARACTER SET LATIN	145552.00000	91479.00000	1.00000	0.00000	null	null	null
3	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonadelavanderia	VARCHAR(2) CHARACTER SET LATIN	145552.00000	46704.00000	1.00000	0.00000	null	null	null
4	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zona_de_bbq	VARCHAR(2) CHARACTER SET LATIN	145552.00000	120685.00000	1.00000	0.00000	null	null	null
5	ADLSLSAMER_MS_AZ	Precio_Casas_Col	vista	VARCHAR(10) CHARACTER SET LATIN	145552.00000	66782.00000	2.00000	0.00000	null	null	null
6	ADLSLSAMER_MS_AZ	Precio_Casas_Col	vigilancia	VARCHAR(10) CHARACTER SET LATIN	145552.00000	46454.00000	3.00000	0.00000	null	null	null
7	ADLSLSAMER_MS_AZ	Precio_Casas_Col	valorventa	NUMBER(38,2)	145552.00000	0.00000	3384.00000	null	2.00000	145550.00000	0.00000
8	ADLSLSAMER_MS_AZ	Precio_Casas_Col	valor	NUMBER(38,2)	145552.00000	0.00000	3384.00000	null	2.00000	145550.00000	0.00000
9	ADLSLSAMER_MS_AZ	Precio_Casas_Col	tipodegaraje	VARCHAR(15) CHARACTER SET LATIN	145552.00000	74957.00000	4.00000	0.00000	null	null	null
10	ADLSLSAMER_MS_AZ	Precio_Casas_Col	tiempodeconstruido	VARCHAR(20) CHARACTER SET LATIN	145552.00000	0.00000	4.00000	0.00000	null	null	null
11	ADLSLSAMER_MS_AZ	Precio_Casas_Col	terraza	VARCHAR(10) CHARACTER SET LATIN	145552.00000	72280.00000	4.00000	0.00000	null	null	null
12	ADLSLSAMER_MS_AZ	Precio_Casas_Col	sauna_yo_turco	VARCHAR(2) CHARACTER SET LATIN	145552.00000	135416.00000	1.00000	0.00000	null	null	null
13	ADLSLSAMER_MS_AZ	Precio_Casas_Col	saloncomunal	VARCHAR(2) CHARACTER SET LATIN	145552.00000	64681.00000	1.00000	0.00000	null	null	null
14	ADLSLSAMER_MS_AZ	Precio_Casas_Col	remodelado	VARCHAR(2) CHARACTER SET LATIN	145552.00000	0.00000	2.00000	0.00000	null	null	null
15	ADLSLSAMER_MS_AZ	Precio_Casas_Col	porteriaovigilancia	VARCHAR(10) CHARACTER SET LATIN	145552.00000	55760.00000	3.00000	0.00000	null	null	null
16	ADLSLSAMER_MS_AZ	Precio_Casas_Col	plantaelectrica	VARCHAR(2) CHARACTER SET LATIN	145552.00000	143274.00000	1.00000	0.00000	null	null	null
17	ADLSLSAMER_MS_AZ	Precio_Casas_Col	piscina	VARCHAR(2) CHARACTER SET LATIN	145552.00000	133205.00000	1.00000	0.00000	null	null	null
18	ADLSLSAMER_MS_AZ	Precio_Casas_Col	parqueaderovisitantes	VARCHAR(2) CHARACTER SET LATIN	145552.00000	127987.00000	1.00000	0.00000	null	null	null
19	ADLSLSAMER_MS_AZ	Precio_Casas_Col	numeroascensores	SMALLINT	145552.00000	61917.00000	4.00000	null	0.00000	83635.00000	0.00000

Step 3. Data Preparation
DataExplorer Syntax


```
call td_analyze('dataexplorer’,
'database = val_source;
Tablename = customer;
Outputdatabase = val_results;
Optional Parameters;');
```

Step 3. Data Preparation
Lab 02b: Exploring the Dataset

teradata.


```
-- Exploring the Values in the dataset
call TRNG_XSP.td_analyze (
'values',
'database = ADLSLSAMER_MS_AZ;
tablename = Precio_Casas_Col;
columns = all;');
```

- We can see the count of nulls, unique values, positive, negative, zeros, etc.

	xdb	xtbl	xcol	xtype	xcnt	xnull	xunique	xblank	xzero	xpos	xneg
1	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonasverdes	VARCHAR(2) CHARACTER SET LATIN	145552.00000	95643.00000	1.00000	0.00000	null	null	null
2	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonaninos	VARCHAR(2) CHARACTER SET LATIN	145552.00000	91479.00000	1.00000	0.00000	null	null	null
3	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zonadelavanderia	VARCHAR(2) CHARACTER SET LATIN	145552.00000	46704.00000	1.00000	0.00000	null	null	null
4	ADLSLSAMER_MS_AZ	Precio_Casas_Col	zona_de_bbq	VARCHAR(2) CHARACTER SET LATIN	145552.00000	120685.00000	1.00000	0.00000	null	null	null
5	ADLSLSAMER_MS_AZ	Precio_Casas_Col	vista	VARCHAR(10) CHARACTER SET LATIN	145552.00000	66782.00000	2.00000	0.00000	null	null	null
6	ADLSLSAMER_MS_AZ	Precio_Casas_Col	vigilancia	VARCHAR(10) CHARACTER SET LATIN	145552.00000	46454.00000	3.00000	0.00000	null	null	null
7	ADLSLSAMER_MS_AZ	Precio_Casas_Col	valorventa	NUMBER(38,2)	145552.00000	0.00000	3384.00000	null	2.00000	145550.00000	0.00000
8	ADLSLSAMER_MS_AZ	Precio_Casas_Col	valor	NUMBER(38,2)	145552.00000	0.00000	3384.00000	null	2.00000	145550.00000	0.00000
9	ADLSLSAMER_MS_AZ	Precio_Casas_Col	tipodegaraje	VARCHAR(15) CHARACTER SET LATIN	145552.00000	74957.00000	4.00000	0.00000	null	null	null
10	ADLSLSAMER_MS_AZ	Precio_Casas_Col	tiempodeconstruido	VARCHAR(20) CHARACTER SET LATIN	145552.00000	0.00000	4.00000	0.00000	null	null	null
11	ADLSLSAMER_MS_AZ	Precio_Casas_Col	terraza	VARCHAR(10) CHARACTER SET LATIN	145552.00000	72280.00000	4.00000	0.00000	null	null	null
12	ADLSLSAMER_MS_AZ	Precio_Casas_Col	sauna_yo_turco	VARCHAR(2) CHARACTER SET LATIN	145552.00000	135416.00000	1.00000	0.00000	null	null	null
13	ADLSLSAMER_MS_AZ	Precio_Casas_Col	saloncomunal	VARCHAR(2) CHARACTER SET LATIN	145552.00000	64681.00000	1.00000	0.00000	null	null	null
14	ADLSLSAMER_MS_AZ	Precio_Casas_Col	remodelado	VARCHAR(2) CHARACTER SET LATIN	145552.00000	0.00000	2.00000	0.00000	null	null	null
15	ADLSLSAMER_MS_AZ	Precio_Casas_Col	porteriaovigilancia	VARCHAR(10) CHARACTER SET LATIN	145552.00000	55760.00000	3.00000	0.00000	null	null	null
16	ADLSLSAMER_MS_AZ	Precio_Casas_Col	plantaelectrica	VARCHAR(2) CHARACTER SET LATIN	145552.00000	143274.00000	1.00000	0.00000	null	null	null
17	ADLSLSAMER_MS_AZ	Precio_Casas_Col	piscina	VARCHAR(2) CHARACTER SET LATIN	145552.00000	133205.00000	1.00000	0.00000	null	null	null
18	ADLSLSAMER_MS_AZ	Precio_Casas_Col	parqueaderovisitantes	VARCHAR(2) CHARACTER SET LATIN	145552.00000	127987.00000	1.00000	0.00000	null	null	null
19	ADLSLSAMER_MS_AZ	Precio_Casas_Col	numeroascensores	SMALLINT	145552.00000	61917.00000	4.00000	null	0.00000	83635.00000	0.00000

Step 3. Data Preparation
Transforming Data

- We will now use the **Vartran** function to prepare data, so it is ready for further analysis in other functions
- Note these functions only work on columns whose data type is numeric

The Variable Transformation functions include:

- Bin Code
- Derive
- Design Code
- Null Replacement
- Recode
- Rescale
- Retain
- Sigmoid
- Z-Score

Step 3. Data Preparation
Why Transform?

- Sometimes the variables have very skewed distributions. To get linear relationships is necessary to transform them.
- There are many possible transformations. The best option is to test various methods to find the better one.

Potencia	Transformación	Descripción
$\lambda_1 = 2$	$Y' = Y^2$	Cuadrado
$\lambda_1 = 1$	$Y' = Y$	Datos sin Transformar
$\lambda_1 = 0.5$	$Y' = \sqrt{Y}$	Raíz Cuadrada
$\lambda_1 = 0.333$	$Y' = \sqrt[3]{Y}$	Raíz Cúbica
$\lambda_1 = 0$	$Y' = \ln(Y)$	Logaritmo
$\lambda_1 = -0.5$	$Y' = \frac{1}{\sqrt{Y}}$	Raíz Cuadrada Inversa
$\lambda_1 = -1$	$Y' = \frac{1}{Y}$	Reciproco

Step 3. Data Preparation
Lab 03a: Transform and Recode


```
call TRNG_XSP.td_analyze('vartran',
'database=ADLSLSAMER_MS_AZ;
tablename=Precio_Casas_Col;
outputstyle=table;
outputdatabase=ADLSLSAMER_MS_AZ;
outputtablename=trans;
keycolumns=id;
index=id;
designcode={designstyle(dummycode),designvalues(Entre 10 y 20
años/ant10_20,Entre 0 y 5 años/ant0_5,Entre 5 y 10 años/ant5_10,
Más de 20 años/ant20_mas),columns(antiguedad_original)};
derive={formula(''sqrt(x)''),arguments(valor),outputname(rvalor)};');
```

- We can use this option to transform features

Step 3. Data Preparation
Lab 3b: Transform, Recode, Impute, Filter and Sampling


```
CREATE TABLE precios AS (
select id, area, habitaciones,
CASE WHEN antiguedad_original='Entre 0 y 5 años' THEN 1 ELSE 0 END AS ant0_5,
CASE WHEN antiguedad_original='Entre 5 y 10 años' THEN 1 ELSE 0 END AS ant5_10,
CASE WHEN antiguedad_original='Entre 10 y 20 años' THEN 1 ELSE 0 END AS ant10_20,
CASE WHEN antiguedad_original='Más de 20 años' THEN 1 ELSE 0 END AS ant20_mas,
CASE WHEN antiguedad_original='Menos de 1 año' THEN 1 ELSE 0 END AS ant1_menos,
CASE WHEN antiguedad_original='1 a 8 años' THEN 1 ELSE 0 END AS ant1_8,
CASE WHEN antiguedad_original='9 a 15 años' THEN 1 ELSE 0 END AS ant9_15,
CASE WHEN antiguedad_original='16 a 30 años' THEN 1 ELSE 0 END AS ant16_30,
CASE WHEN antiguedad_original='Más de 30 años' THEN 1 ELSE 0 END AS ant30_mas,
CASE WHEN banos is null then 0 else banos end as banos,
CASE WHEN garajes is null then 0 else garajes end as garajes,
CASE WHEN estrato is null then 0 else estrato end as estrato, SQRT(valor) as rvalor,
SAMPLEID as sid
FROM Precio_Casas_Col
WHERE area between 20 and 2000 and valor between 50000000 and 5000000000
SAMPLE RANDOMIZED ALLOCATION 0.7, 0.3
) WITH DATA
PRIMARY INDEX (id);
```

Sometimes, single SQL code is useful to implement various transformations

Step 4. Modeling

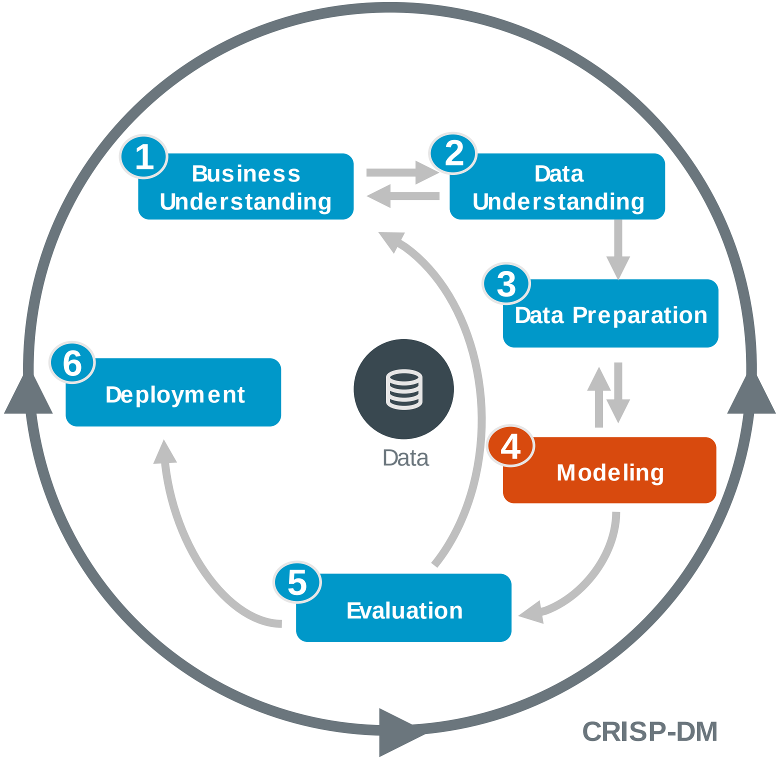
a) If ML, which Function (algorithms) and which order of execution?

Linear: (Linear Regression Model) Formulates a model by discovering the linear relationships between independent and dependent variables.

LinearScore: Uses the previous output to predict the dependent variable against unknown data

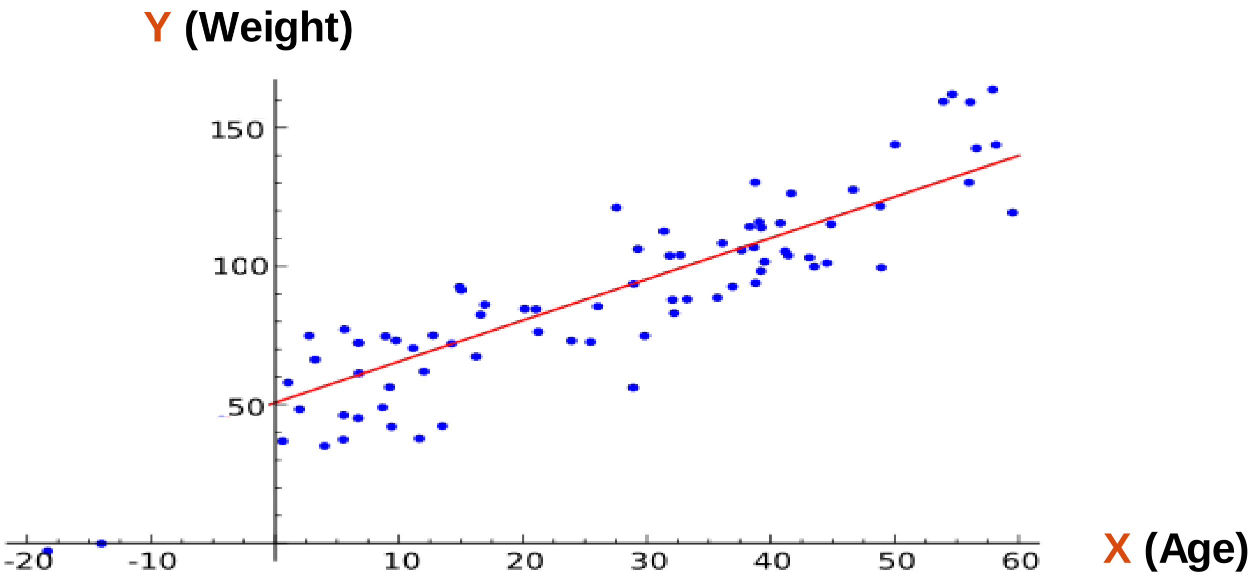
b) Does data require more Cleaning for the function to process?

No



Step 4. Modeling
Linear Description

The **Linear Regression Model** enables the linear equation to relate to the dependent variables by a Link function.



Formula for Predicting Y:

$Y = a + bX$

Y = Value being predicted (Weight)

a = Y-intercept (50)

b = Slope of line (.3)

X = Value you know (Age)

Note: The line is fit such that the sum of all the squared residuals (distances between each actual datapoint's y-value and the line's same y-value) is as small as possible

In Cause and Effect, the Independent variable (X) is the Cause and the Dependent variable (Y) is the Effect

Step 4. Modeling
Lab 04: Create TRAIN and TEST Tables


```sql
-- SAMPLE  data into 2 tables
CREATE MULTISET TABLE tch_train AS (
SELECT * FROM precios WHERE SID = 1
) WITH DATA
PRIMARY INDEX (id);

CREATE MULTISET TABLE tch_test AS (
SELECT * FROM precios WHERE SID = 2
) WITH DATA
PRIMARY INDEX (id);
```

- We create a TRAIN and TEST table from the precios table (so Outliers have been resolved and Scaling performed)
- We will use the **Linear** function to build a Model for predicting rvalor (the square root of the published price for a house)

	id	area	habitacio...	ant10_20	ant0_5	ant5_10	ant20_mas	ant1_8	ant16_30	ant9_15	ant30_mas	ant1_menos	banos	garajes	estrat	rvalor	sid
1	41089	305.00000	3	0	0	0	0	1	0	0	0	0	4	4	6	54772.2557	1
2	129324	84.00000	2	0	0	1	0	0	0	0	0	0	2	2	5	21447.6105	1
3	39171	53.00000	3	0	0	0	0	0	0	0	1	0	1	1	3	12884.0987	1
4	84014	55.00000	3	1	0	0	0	0	0	0	0	0	1	0	3	12206.5556	1
5	37133	112.00000	2	0	1	0	0	0	0	0	0	0	3	1	4	25000.0000	2
6	2793	117.00000	3	0	0	0	1	0	0	0	0	0	2	1	5	20976.1769	2
7	57790	71.00000	2	1	0	0	0	0	0	0	0	0	2	1	4	15811.3883	1
8	41558	65.50000	2	0	1	0	0	0	0	0	0	0	2	1	5	20116.2372	1
9	116947	325.00000	3	1	0	0	0	0	0	0	0	0	5	4	5	52915.0262	1
10	57321	125.00000	3	0	0	0	0	0	1	0	0	0	3	2	4	25495.0975	1

Step 4. Modeling
Linear Syntax


```
call td_analyze('linear',
'database = val_database;
Tablename = tablename;
Columns = c1, c2;
Dependent = c0;
Groupby = c3, c4;
Outputdatabase = val_out_db;
Outputtablename = val_out_tbl;
Constant = true;
Optional Parameters;');
```

Step 4. Modeling
Lab 05: Create Model

teradata.


```
call TRNG_XSP.td_analyze('linear',
'database=ADLSLSAMER_MS_AZ;
tablename=tch_train;
columns= all;
columnstoexclude=id,sid;
dependent=rvalor;
statstable=true;
stepwise=true;
outputdatabase=ADLSLSAMER_MS_AZ;
outputtablename=linearmodel;');
```

	Column Name	B Coefficient	Standard Error	T Statistic	P-Value
1	(Constant)	7773.21359	116.16625	66.91456	0.00000
2	ant0_5	623.36157	105.50324	5.90846	0.00000
3	ant1_8	-412.72735	112.10057	-3.68176	0.00023
4	ant1_menos	642.28020	143.20291	4.48511	0.00001
5	ant10_20	-1266.36810	104.88542	-12.07382	0.00000
6	ant16_30	-1951.35734	113.82492	-17.14350	0.00000
7	ant20_mas	-1687.62400	109.91202	-15.35432	0.00000
8	ant30_mas	-1315.23379	140.96906	-9.32995	0.00000
9	ant5_10	-509.27420	106.53567	-4.78032	0.00000
10	ant9_15	-1224.16156	116.80606	-10.48029	0.00000
11	area	79.64339	0.27095	293.94481	0.00000
12	banos	1399.47085	19.30932	72.47645	0.00000
13	estrato	794.48328	9.91947	80.09332	0.00000
14	garajes	2222.22479	20.37773	109.05163	0.00000
15	habitaciones	-1002.33398	17.88066	-56.05688	0.00000

The Dependent column must be of a numeric data type

Step 4. Modeling
LinearScore Syntax

teradata.


```
call td_analyze('linearscore',
'database = db;
Tablename = tbl;
Outputdatabase = out_db;
Outputtablename = out_tbl;
Modeldatabase = model_db;
Modeltablename = model_tbl;
Index = i1, i2, i3;
Retain = r1, r2, r3;
Scoringmethod = { score| evaluate| scoreandevaluate};
Residual = res;
Predicted = pre;
Optional Parameters;');
```

Note: The function may fail if you have NULL values present in your input table

Step 4. Modeling
Lab 06: LinearScore


```
call TRNG_XSP.td_analyze('linearscore',
'database=ADLSLSAMER_MS_AZ;
tablename=tch_test;
modeldatabase=ADLSLSAMER_MS_AZ;
modeltablename=linearmodel;
outputdatabase=ADLSLSAMER_MS_AZ;
outputtablename=linearmodelval;
predicted=valor_estimado;
retain=valor;
samplescoresize=10;
scoringmethod=scoreandevaluate;');
```

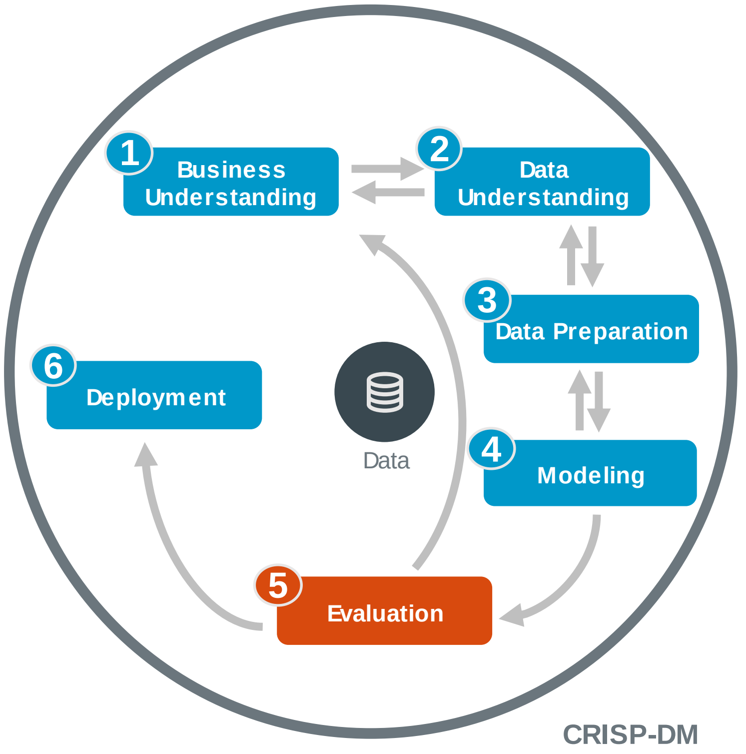
		Actual	Prediction	
	id	rvalor	rvalor_estimado	Residual
1	107882	14832.39697	16838.37395	-2005.97697
2	70976	12449.89960	11026.00345	1423.89615
3	4895	27568.09750	26787.33594	780.76157
4	10408	23600.84744	24457.04617	-856.19873
5	42247	19104.97317	20166.04391	-1061.07074
6	71517	18708.28693	20053.37047	-1345.08354
7	13180	30000.00000	34377.27935	-4377.27935
8	84973	20663.97832	20322.50958	341.46874
9	140887	20248.45673	20218.83499	29.62174
10	103066	27018.51217	26268.18507	750.32710

Step 5. Evaluation

5. Evaluation – Assessing the business value of the model/analysis output

   a) Is Output Actionable (sufficiently meet business goals/success criteria), nice to know or new starting point?

Let's Evaluate the Output…



Step 5. Evaluation
Regression Model Output – Is It Actionable?

- Our Model does a very poor job at meeting our stated business goal and predicting the chosen dependent variable
- Next steps could include one or more of the following:
  - Re-confirm that source data is complete and correct
  - Re-run the model with different arguments
  - Re-run the model without removing outliers
  - Re-run the model without scaling the data
  - Build other models using different predictive functions
  - Re-assess how important the stated business goal actually is

Total Obser...	Total Sum of Squares	Multiple Co...	Squared Mul...	Adjusted R-Sq...	Standard Error o...	Regression Sum of ...	Regression Degr...	Regression Mean-...	Regression F Ratio	Regression P-Va...
100779.00000	9039795501953.05000	0.90843	0.82525	0.82522	3959.46802	7460079280549.98000	14.00000	532862805753.57000	33989.26151	0.00000

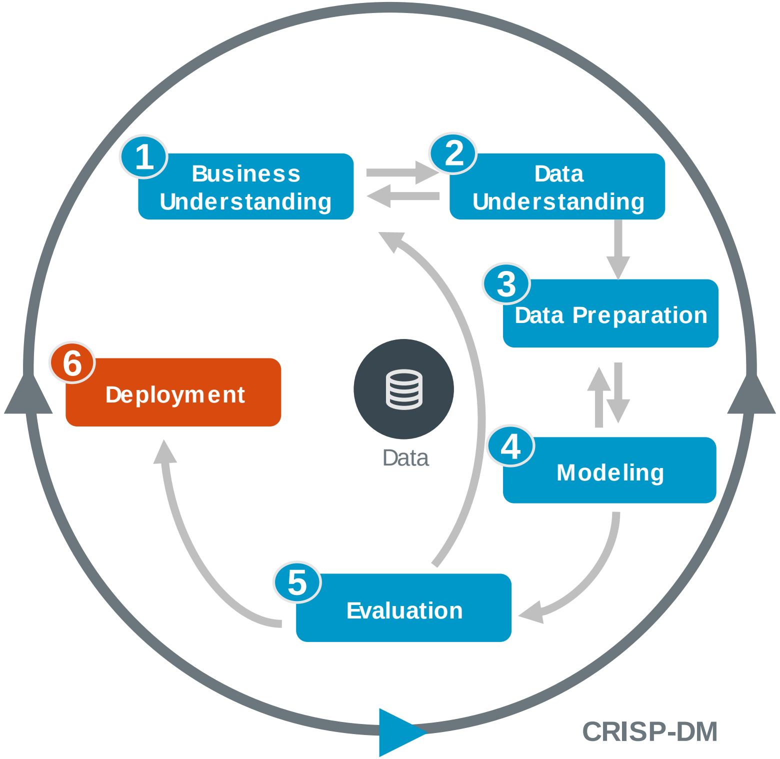
Step 6a. Deployment (Operationalizing/Monitoring/Maintenance)

teradata.

6. **Deployment** – The end goal is to "operationalize" the analytic findings. Taking analytics from insight to impact – the process of getting analytics out to business stakeholders for use/reuse to meet business goals

   a) **Plan deployment (how to operationalize)**

   Note: This varies by customer and is not covered in this course



CRISP-DM

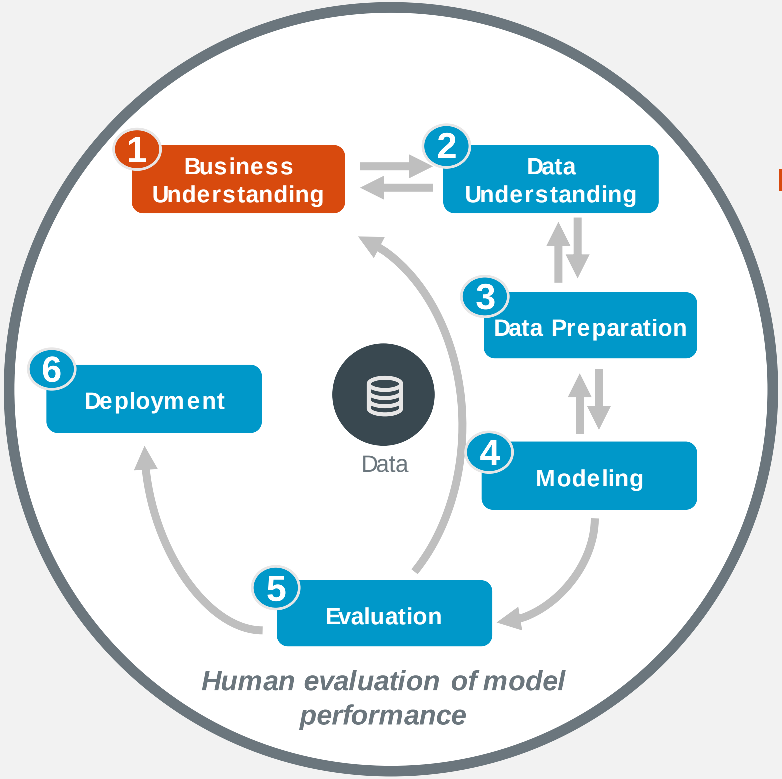
Current Topic – House Pricing Model with Data Science Process (Python)

teradata.

- Introduction
- **House Pricing Model**
  - **Data Science Process**
    - SQL
    - **Python**
    - R
- Summary and Review

Step 1. Business Understanding
Map Out Data Science Process – Tools and Functions

Business Objective:
Build a Model that accurately predicts House Prices. Prepare data first



CRISP-DM

Language: Python in JupyterLab

Data Preparation Functions: Values, Statistics, Transform

Modeling Functions: Linear, LinearScore

Visualization apps: N/A

Open JupyterLab

1. Open file: Demo_Price_Model_Python.ipynb
2. Highlight the 1st Cell (you'll get a blue vertical bar for that Cell)

Modelo de Estimación del Precio de Viviendas

Step 1: Business Understanding

Este dataset contiene información recolectada de precios y características de 142 mil viviendas en Colombia. La información se encuentra disponible públicamente en el repositorio Kaggle: https://www.kaggle.com/datasets/danieleduardofajardo/colombia-house-prediction

Cargamos las librerías

[1]:

```
from teradataml import create_context, DataFrame, get_context, copy_to_sql, in_schema, remove_context
from teradataml.dataframe.sql_functions import case
import tdconnect
import pandas as pd
import numpy as np
import getpass as gp

from teradataml import *
from teradataml.analytics.valib import *
configure.val_install_location = "TRNG_XSP"
```

Lab 00: Import Python Libraries

1. Highlight Cell 1 (you'll get a blue vertical bar for that Cell)
2. Click Run button ▶. Kernel indicator circle will fill in. When finished it will be White again (sometimes it happens so fast won't see circle fill in)
3. Run Cell 2 to Display all Python code as SQL code

Cargamos las librerías

```
[1]: from teradataml import create_context, DataFrame, get_context, copy_to_sql, in_schema, remove_context
from teradataml.dataframe.sql_functions import case
import tdconnect
import pandas as pd
import numpy as np
import getpass as gp

from teradataml import *
from teradataml.analytics.valib import *
configure.val_install_location = "TRNG_XSP"
```

Step 2. Data Understanding
Lab 01: Load/View the Raw Data

Business Objective: Build a Model to predict House Prices. Prepare data by first modifying Outlier values and Normalizing the data

```
tdPrecios = DataFrame(in_schema("ADLSLSAMER_MS_AZ", "Precio_Casas_Col"))
tdPrecios.to_pandas().head(10)
```

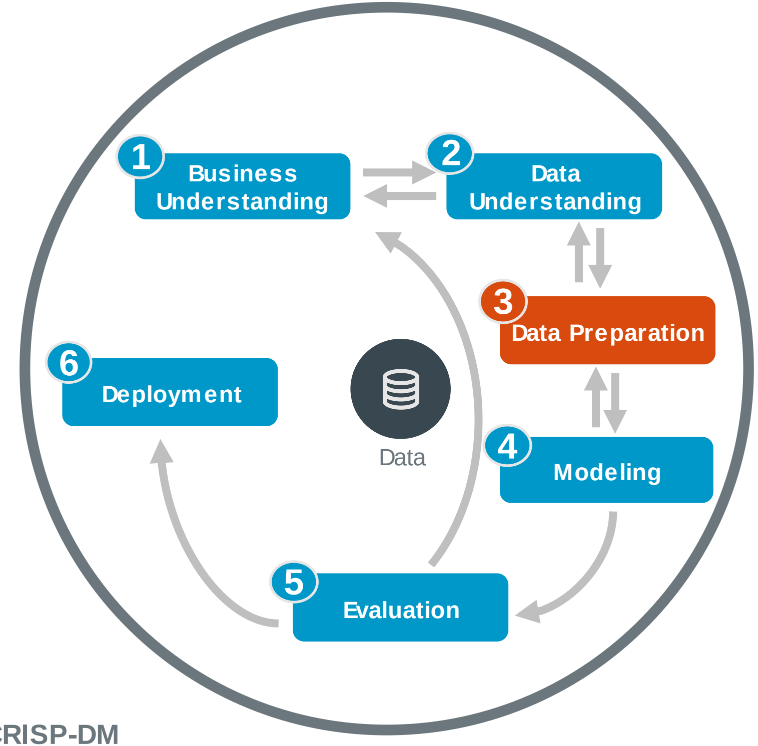
id	antiguedad_original	area	areabalcon	areaconstruida	areaterraza	balcon	banos	banoservicio	conjuntocerrado	cuarto_de_escoltas	...	tiempodeconstruido	tipodegaraje	valor
112194	Entre 10 y 20 años	179	None	179	None	Ninguno	3.0	None	None	None	...	Entre 10 y 20 años	None	1150000000
88784	Entre 0 y 5 años	65	None	65	None	Balcón	2.0	None	None	None	...	Entre 0 y 5 años	Independiente	560500000
60011	Entre 0 y 5 años	38	None	38	None	Ninguno	1.0	None	Si	None	...	Entre 0 y 5 años	None	165000000
2669	Entre 0 y 5 años	124	32	124	32	Terraza	3.0	None	Si	None	...	Entre 0 y 5 años	Servidumbre	855000000
44921	Entre 10 y 20 años	61	None	61	None	Terraza	2.0	None	None	None	...	Entre 10 y 20 años	None	310000000
70207	Entre 0 y 5 años	62	None	62	None	Balcón	2.0	None	Si	None	...	Entre 0 y 5 años	None	330000000
115681	Más de 20 años	97	None	97	None	None	3.0	None	Si	None	...	Más de 20 años	Independiente	320000000
30565	Entre 10 y 20 años	257	None	257	None	Ninguno	3.0	None	Si	None	...	Entre 10 y 20 años	None	1800000000
85297	Entre 10 y 20 años	57	None	57	None	None	1.0	None	None	None	...	Entre 10 y 20 años	None	195000000
30769	1 a 8 años	97	None	97	None	None	2.0	None	None	None	...	Entre 0 y 5 años	None	599000000

- Describe and explore data — Table houses various characteristics related to houses. The id is the unique identifier. Our dependent variable is valor
- Verify data adequacy/quality — Yes, data is adequate for analysis task (Complete/Correct)

Step 3. Data Preparation

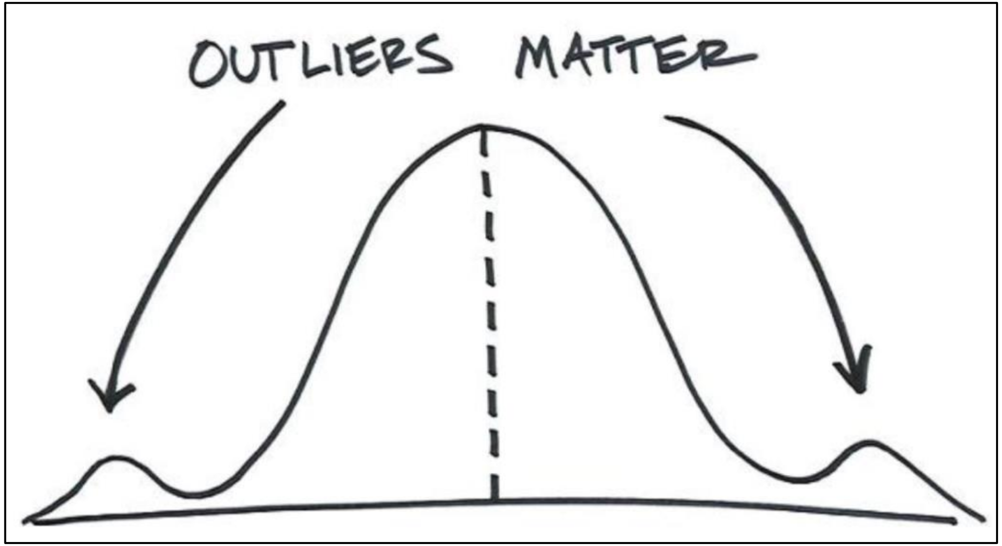
a) Does data require Cleaning? Does data need to be Scaled? Do Outliers need to be removed?

Yes. We will remove Outliers and Scale desired variables



CRISP-DM

Step 3. Data Preparation
Identify Outliers



- Detecting and handling outliers becomes critical for certain statistical measures
- Analysis of data may not be meaningful if outliers are in the data

The **DataExplorer, Values, Statistics** options might be useful to detect abnormal and missing values; e.g., is could be useful for identifying the very best of your customers, it could be used to detect fraudulent activity, etc.

Step 3. Data Preparation
Lab 02a: Values

teradata.

```
explor = valib.Values(data=tdPrecios,
columns="all")
explor.result.to_pandas()
```

To identify nulls, positive, negative, missing values, etc.

```
explor = valib.Values(data=tdPrecios, columns="all")
explor.result.to_pandas()
```

xdb	xtbl	xcol	xtype	xcnt	xnull	xunique	xblank	xzero	xpos	xneg
ADLSLSAMER_MS_AZ	Precio_Casas_Col	vista	VARCHAR(10) CHARACTER SET LATIN	145552.0	66782.0	2.0	0.0	NaN	NaN	NaN
		id	INTEGER	145552.0	0.0	145552.0	NaN	1.0	145551.0	0.0
		garajes	SMALLINT	145552.0	13864.0	9.0	NaN	0.0	131688.0	0.0
		saloncomunal	VARCHAR(2) CHARACTER SET LATIN	145552.0	64681.0	1.0	0.0	NaN	NaN	NaN
		areabalcon	NUMBER(8,2)	145552.0	121195.0	336.0	NaN	11.0	24346.0	0.0
		porteriaovigilancia	VARCHAR(10) CHARACTER SET LATIN	145552.0	55760.0	3.0	0.0	NaN	NaN	NaN
		estrato	SMALLINT	145552.0	6667.0	7.0	NaN	0.0	138885.0	0.0
		cuartodeservicio	VARCHAR(2) CHARACTER SET LATIN	145552.0	117209.0	1.0	0.0	NaN	NaN	NaN
		depositoocuartoutil	VARCHAR(10) CHARACTER SET LATIN	145552.0	88096.0	4.0	0.0	NaN	NaN	NaN
		numeroascensores	SMALLINT	145552.0	61917.0	4.0	NaN	0.0	83635.0	0.0

Step 3. Data Preparation
Lab 02b: Statistics

teradata.

```
out = valib.Statistics(data=tdPrecios,
columns=["area", "valor"],
extended_options="quantiles")
out.result.to_pandas()
```

To detect Outliers with Percentile Option

```
out = valib.Statistics(data=tdPrecios, columns=["area", "valor"], extended_options="quantiles")
out.result.to_pandas()
```

xcol	xcnt	xmin	xmax	xmean	xstd	xpctile0	xpctile1	xpctile2	xpctile3	xpctile4	...	xpctile91	xpctile92	xpctile93	xpctile94	xpctile95
valor	145552.0	0.0	4.100000e+12	1.401152e+09	2.829426e+10	0.0	115000000.0	130000000.0	145000000.0	150000000.0	...	1.478000e+09	1.550000e+09	1.600000e+09	1.750000e+09	1.900000e+09
area	145528.0	0.0	5.700000e+05	1.370333e+02	2.130737e+03	0.0	32.0	38.0	41.0	44.0	...	2.140000e+02	2.250000e+02	2.380000e+02	2.500000e+02	2.650000e+02

Step 3. Data Preparation
Lab 03a: Transforming with SQL


```
ndf = DataFrame.from_query("select id, area, habitaciones,
CASE WHEN antiguedad_original='Entre 10 y 20 años' THEN 1 ELSE 0 END AS ant10_20,
CASE WHEN antiguedad_original='Entre 0 y 5 años' THEN 1 ELSE 0 END AS ant0_5,
CASE WHEN antiguedad_original='Entre 5 y 10 años' THEN 1 ELSE 0 END AS ant5_10,
CASE WHEN antiguedad_original='Más de 20 años' THEN 1 ELSE 0 END AS ant20_mas,
CASE WHEN antiguedad_original='1 a 8 años' THEN 1 ELSE 0 END AS ant1_8,
CASE WHEN antiguedad_original='16 a 30 años' THEN 1 ELSE 0 END AS ant16_30,
CASE WHEN antiguedad_original='9 a 15 años' THEN 1 ELSE 0 END AS ant9_15,
CASE WHEN antiguedad_original='Más de 30 años' THEN 1 ELSE 0 END AS ant30_mas,
CASE WHEN antiguedad_original='Menos de 1 año' THEN 1 ELSE 0 END AS ant1_menos,
banos, garajes, estrato, valor, SAMPLEID as sid
FROM ADLSLSAMER_MS_AZ.Precio_Casas_Col
WHERE area between 20 and 2000 and valor between 50000000 and 5000000000
SAMPLE RANDOMIZED ALLOCATION 0.7, 0.3", True, "id")
```

- Is it possible to build a query and store it on a TeradataML Data Frame
- It helps to simplify the transformation process

Step 3. Data Preparation
Lab 03b: Transformations with ‘Transform’


```
fn_1 = FillNa(style="literal", value=0, columns="garajes")
fn_2 = FillNa(style="literal", value=0, columns="estrato")
fn_3 = FillNa(style="literal", value=0, columns="banos")
```

```
derive = Derive(formula="sqrt(x)", columns="valor", out_column="rvalor")
```

```
retain = Retain(columns=["habitaciones",
"area","ant0_5","ant5_10","ant10_20","ant20_mas","ant1_menos","ant1_8",
"ant9_15","ant16_30","ant30_mas","sid"])
```

```
matriz = valib.Transform(data=ndf, fillna=[fn_1, fn_2, fn_3], derive=derive,
retain=retain, key_columns="id", index_columns="id")
```

Step 4. Modeling

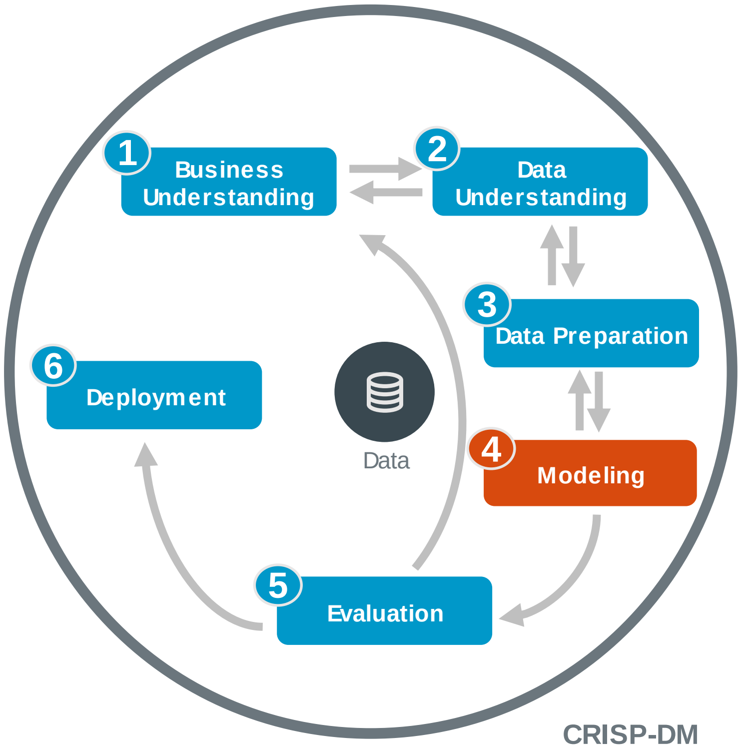
a) If ML, which Function (algorithms) and which order of execution?

Linear: (Linear Regression Model) Formulates a model by discovering the linear relationships between independent and dependent variables.

LinearScore: Uses the previous output to predict the dependent variable against unknown data

b) Does data require more Cleaning for the function to process?

No



CRISP-DM

Step 2. Data Understanding
Lab 03: View Prepped Dataframes


```
tbl_train=matriz.result[matriz.result["sid"]==1]

tbl_test=matriz.result[matriz.result["sid"]==2]
```

- Here, we view our TRAIN set and TEST Dataframes
- We will use the Linear Regression algorithm to build a Model to predict the House Prices

Step 4. Modeling
Lab 04: Create the Model


```
tdModel = valib.LinReg(data=tbl_train,
                       columns="all",
                       exclude_columns=["id", "sid"],
                       stepwise=True,
                       response_column="rvalor")
tdModel.model.to_pandas()
```

- We can use feature selection algorithms, despite Scikit-Learn

The ResponseColumn must be of a numeric data type

Column Name	B Coefficient	Standard Error	T Statistic	P-Value	Lower	Upper	Standard Coefficient	Incremental R-Squared
ant10_20	-1252.157310	104.279709	-12.007679	3.411702e-33	-1456.544239	-1047.770381	-0.058506	0.823224
ant16_30	-1976.882952	113.547119	-17.410243	8.674452e-68	-2199.433890	-1754.332014	-0.049794	0.823001
(Constant)	7378.714421	115.801885	63.718431	0.000000e+00	7151.744170	7605.684671	0.000000	0.000000
ant0_5	690.657957	104.981707	6.578841	4.764389e-11	484.895121	896.420792	0.031054	0.811712
ant30_mas	-1293.317418	139.206210	-9.290659	1.562617e-20	-1566.159853	-1020.474983	-0.017852	0.823433
ant1_menos	694.376851	143.804907	4.828603	1.376952e-06	412.521026	976.232676	0.008967	0.822066
ant20_mas	-1653.428861	109.248482	-15.134571	1.092252e-51	-1867.554523	-1439.303199	-0.051083	0.823178
banos	1258.267216	19.254005	65.350933	0.000000e+00	1220.529605	1296.004826	0.138237	0.816933
estrato	821.623526	9.931604	82.728183	0.000000e+00	802.157707	841.089345	0.130373	0.804314
habitaciones	-766.888197	18.150366	-42.251941	0.000000e+00	-802.462689	-731.313706	-0.069330	0.820687
ant1_8	-408.872748	111.596433	-3.663851	2.485782e-04	-627.600365	-190.145131	-0.011265	0.822722
area	78.215542	0.269210	290.537213	0.000000e+00	77.687893	78.743190	0.606413	0.722179
ant9_15	-1202.414236	116.430822	-10.327285	5.458667e-25	-1430.617194	-974.211278	-0.027261	0.823282
garajes	2312.012685	20.469241	112.950583	0.000000e+00	2271.893229	2352.132142	0.218030	0.786930
ant5_10	-459.052575	105.967585	-4.332009	1.478973e-05	-666.747720	-251.357430	-0.018224	0.821456

Step 4. Modeling
Lab 05: Model Metrics

```
tdModel.statistical_measures.to_pandas()
```

```
tdModel.statistical_measures.to_pandas()
```

Total Sum of Squares	Multiple Correlation Coefficient (R):	Squared Multiple Correlation Coefficient (1-Tolerance)	Adjusted R-Squared	Standard Error of Estimate	Regression Sum of Squares	Regression Degrees of Freedom	Regression Mean-Square	Regression F Ratio	Regression P-Value	Residual Sum of Squares	Residual Degrees of Freedom	Residual Mean-Square	Output Database
9.035680e+12	0.907432	0.823433	0.823409	3979.07728	7.440278e+12	14.0	5.314484e+11	33565.752847	0.0	1.595402e+12	100764.0	1.583306e+07	LC250058

- The Adjusted R-Square is 82.34%

Step 4. Modeling/ Step 5. Evaluation
Lab 06: Validate Accuracy


```
valib.LinRegEvaluator(data=tbl_test, model=tdModel.model)
```

Note: We can use the Test Sample to Validate the Model

```
valib.LinRegEvaluator(data=tbl_train, model=tdModel.model)
```

```
############ result Output ############

   Minimum Absolute Error  Maxmum Absolute Error  Average Absolute Error  Standard Error of Estimate
0                0.010003          152315.629281             2500.128297                 3978.836434
```

```
valib.LinRegEvaluator(data=tbl_test, model=tdModel.model)
```

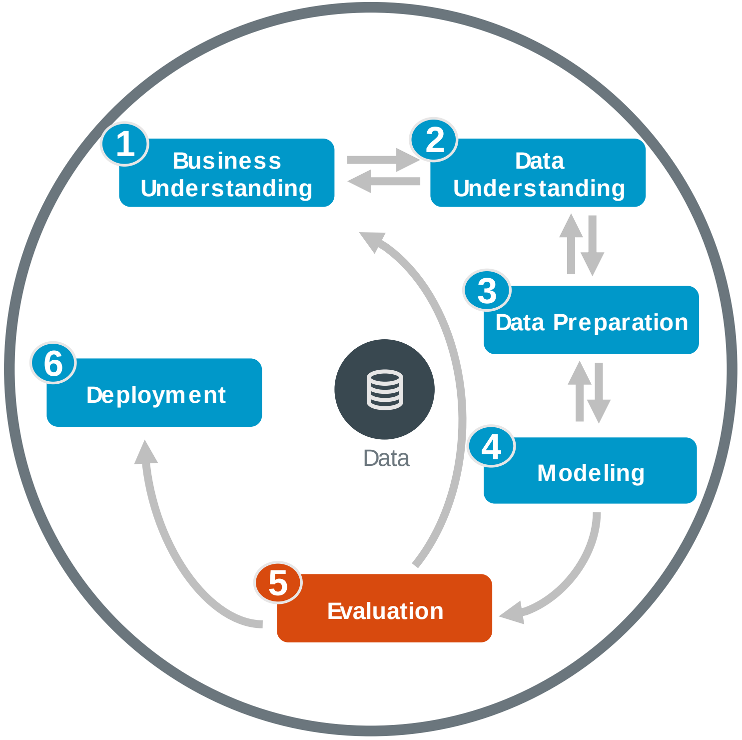
```
############ result Output ############

   Minimum Absolute Error  Maxmum Absolute Error  Average Absolute Error  Standard Error of Estimate
0                0.010003          150234.526457             2513.567227                 3937.149252
```

Step 5. Evaluation

5. Evaluation – Assessing the business value of the model/analysis output

   a) Is Output Actionable (sufficiently meet business goals/success criteria), nice to know or new starting point?

Let's Evaluate the Output…



CRISP-DM

Step 5. Evaluation
Model Output – Is It Actionable?

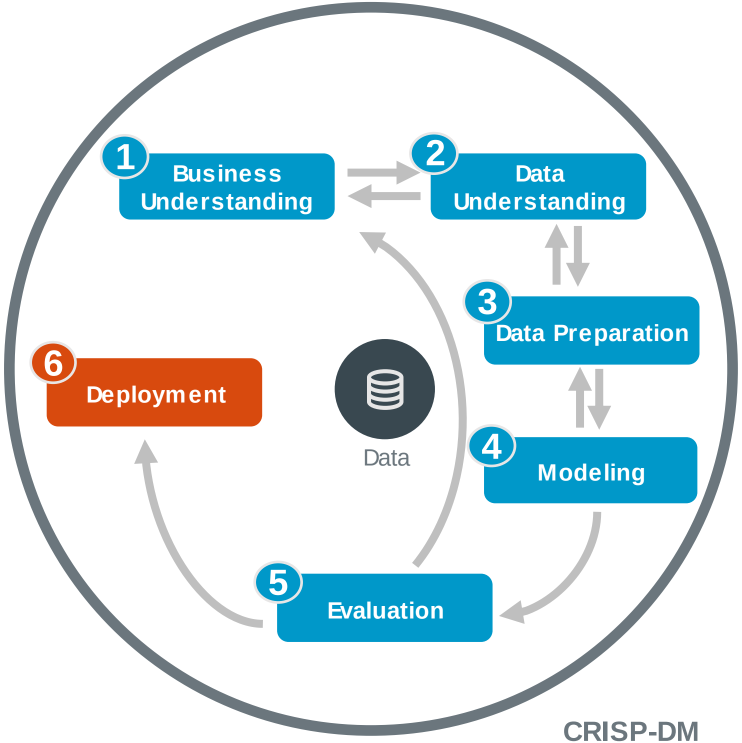
- Our Model does a very poor job at meeting our stated business goal and predicting the chosen dependent variable
- Next steps could include one or more of the following:
  - Re-confirm that source data is complete and correct
  - Re-run the model with different arguments
  - Re-run the model without removing outliers
  - Re-run the model without scaling the data
  - Build other models using different predictive functions
  - Re-assess how important the stated business goal actually is

Step 6a. Deployment (Operationalizing/Monitoring/Maintenance)

6. **Deployment** – The end goal is to "operationalize" the analytic findings. Taking analytics from insight to impact – the process of getting analytics out to business stakeholders for use/reuse to meet business goals

   a) **Plan deployment (how to operationalize)**

   Note: This varies by customer and is not covered in this course



CRISP-DM

Step 6. Deployment
Lab 07: Scoring

LinRegPredict Function

```
tdScore = valib.LinRegPredict(data=tbl_test,
model=tdModel.model, response_column="rvalue_estim")
```

Transform the response into the original scale for Prices

```
derive = Derive(formula="x*x", columns="rvalue_estim",
out_column="valor_estim")
ScoreFinal = valib.Transform(data=tdScore.result,
derive=derive, key_columns="id", index_columns="id")
```

Store the final dataset into Vantage

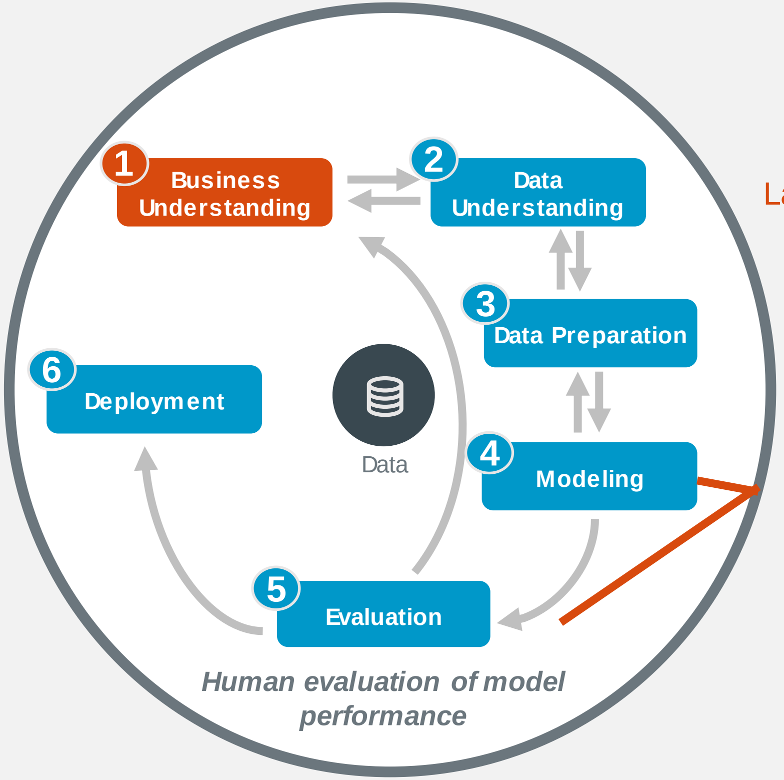
```
ScoreFinal.result.to_sql(schema_name="ADLSLSAMER_MS_AZ",
table_name="Precio_Score")
```

Current Topic – House Pricing Model with Data Science Process (R)

- Introduction
- **House Pricing Model**
  - **Data Science Process**
    - SQL
    - Python
    - **R**
- Summary and Review

Step 1. Business Understanding
Map Out Data Science Process – Tools and Functions

Business Objective:
Build a Model that accurately predicts House Prices. Prepare data first



CRISP-DM

Language: R in RStudio

Data Preparation Functions: td_explore_valib

Modeling Functions: td_lin_reg_valib

Visualization apps: N/A

Lab 00: Load Libraries

Load Dependent R Libraries followed by 'tdplyr'

```
# Load Libraries
LoadPackages <- function() {
library(getPass)
library(dbplyr)
library(DBI)
library(tidyverse)
library(teradatasql)
library(tdplyr)
}

# Suppress Package Detailed Information
"suppressPackageStartupMessages(LoadPackages())
```

Lab 00: Create and Set Teradata Vantage Context

```
# Create Vantage Context
con <- td_create_context (
             host = "host_name",
             uid = "user_id",
             pwd = getpass(),
             dType = "native",
             logmech = "LDAP")

# Connect to Vantage
td_set_context(con)
```

Your code may vary slightly from this Generic example

Create a variable name **con**

1. Use the **td_create_context** function
2. Input the appropriate information for the remaining arguments.

3. Input the **con** variable as the parameter using the **td_set_context** function

Step 2. Data Understanding
Lab 01: Load/View the Raw Data

Business Objective: Build a Model to predict House Prices. Prepare data by first modifying Outlier values and Transforming the data

```
tdPrecios <- tbl(con,dplyr::sql("SELECT * FROM
ADLSLSAMER_MS_AZ.Precio_Casas_Col"))
as.data.frame(head(tdPrecios))
```

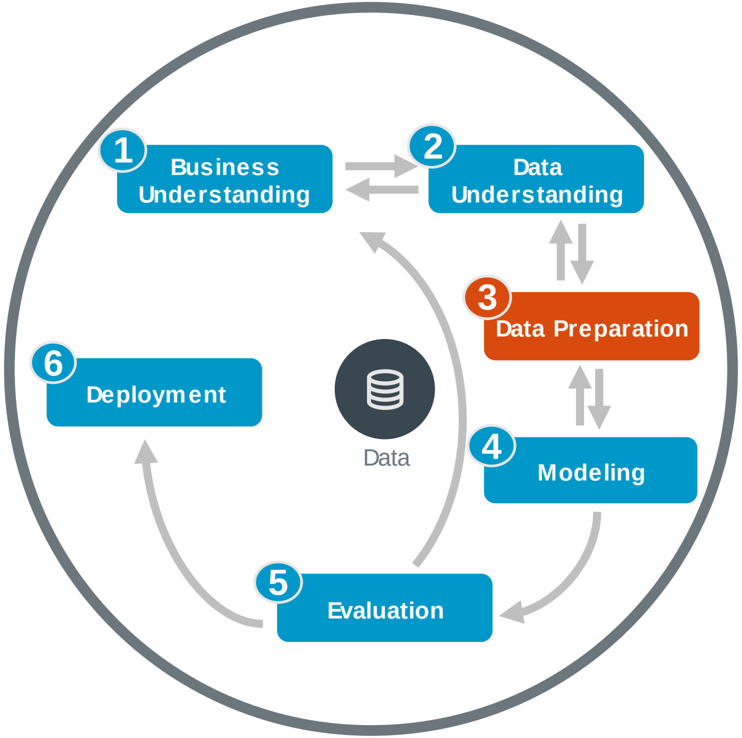
id	area	habitaciones	ant10_20	ant0_5	ant5_10	ant20_mas	ant1_8	ant16_30	ant9_15	ant30_mas	ant1_menos	banos	garajes	estrato	valor	sid
<int>	<dbl>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>
63906	98	3	0	1	0	0	0	0	0	0	0	2	1	NA	4.50e+08	2
11050	44	2	0	0	0	0	0	0	1	0	0	2	NA	3	1.90e+08	1
29566	71	2	0	0	0	1	0	0	0	0	0	2	2	6	4.40e+08	1
133993	60	2	1	0	0	0	0	0	0	0	0	1	1	4	2.60e+08	1
24998	53	3	0	1	0	0	0	0	0	0	0	2	NA	2	1.50e+08	1
23530	67	2	1	0	0	0	0	0	0	0	0	1	1	3	1.85e+08	1

- Describe and explore data — Table houses various characteristics related to houses. The id is the unique identifier. Our dependent variable is valor
- Verify data adequacy/quality — Yes, data is adequate for analysis task (Complete/Correct)

Step 3. Data Preparation

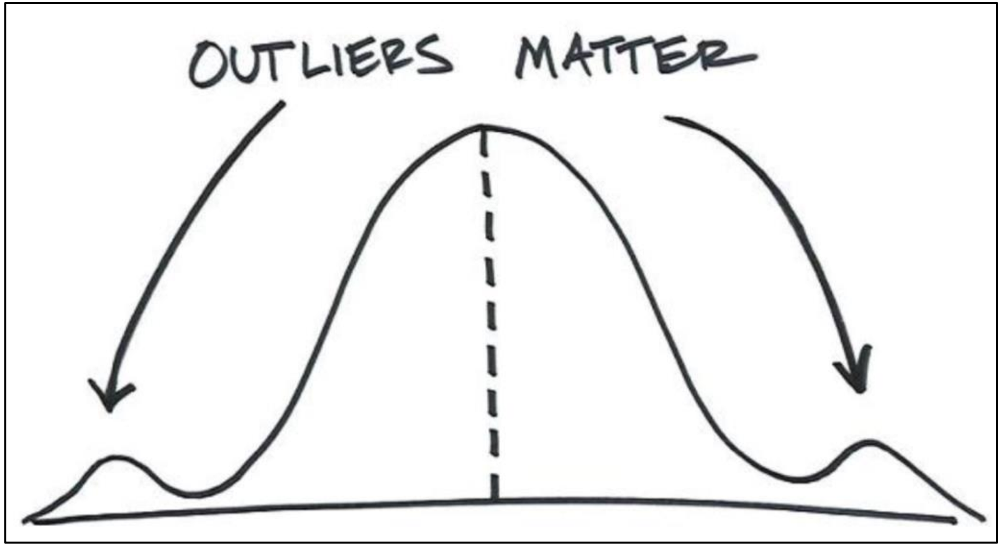
a) Does data require Cleaning? Does data need to be Scaled? Do Outliers need to be removed?

Yes. We will remove outliers and scale desired variables



CRISP-DM

Step 3. Data Preparation
Identify Outliers



- Detecting and handling outliers becomes critical for certain statistical measures
- Analysis of data may not be meaningful if outliers are in the data

The **DataExplorer, Values, Statistics** options might be useful to detect abnormal and missing values; e.g., is could be useful for identifying the very best of your customers, it could be used to detect fraudulent activity, etc.

Step 3. Data Preparation
Lab 02: Explore Features

```
# Run td_explore_valib
eda01 <- td_explore_valib(data=tdPrecios)

# Values
arrange(as.data.frame(eda01$values.output),xcol)

# Statistics
arrange(as.data.frame(eda01$statistics.output),xcol)

# Frequency
arrange(as.data.frame(eda01$frequency.output),xcol,xval)

# Histogram
arrange(as.data.frame(eda01$histogram.output),xcol,xbin)
```

Step 4. Modeling

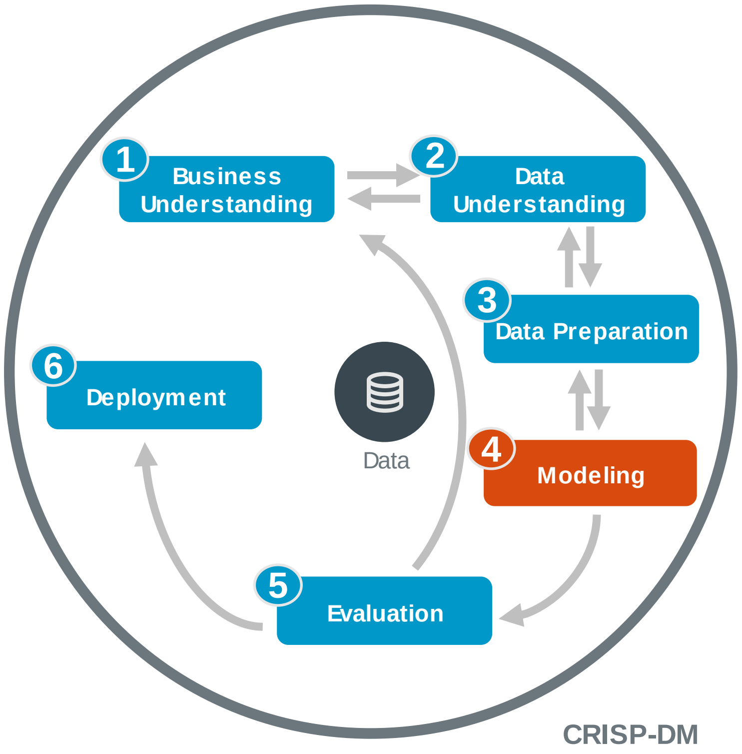
a) If ML, which Function (algorithms) and which order of execution?

Linear: (Linear Regression Model) Formulates a model by discovering the linear relationships between independent and dependent variables.

LinearScore: Uses the previous output to predict the dependent variable against unknown data

b) Does data require more Cleaning for the function to process?

No



Step 4. Modeling
Lab 03: View TRAIN and TEST Tibbles

Business Objective: Build a Model to predict House Prices. But prepare data by first modifying Outlier values and Transforming the data

- We will next run a **Linear** model against the table, which has been pre-populated for you and is based on your previously-created tibble, which had already been treated for Outliers and Transformed
- The same holds true for the model table, for when we run **LinearScore**

```
-- Samples
tbl_train <- filter(tdPrecios, sid == 1)
tbl_test <- filter(tdPrecios, sid == 2)
```

Step 4. Modeling
Lab 04: Create the Model


```
# Training Model with Stepwise Selection
tdModel <- td_lin_reg_valib(data=tbl_train,
                            columns='all',
                            stepwise='True',
                            response.column='valor')
```

The training function for Linear Regression

The ResponseColumn must be of a numeric data type

Step 4. Modeling/Step 5. Evaluation
Lab 05: Validate Accuracy


```
# Model Statistics
print(tdModel$statistical.measures)

# Model Evaluation
tdEval <- td_lin_reg_evaluator_valib(data=tbl_test, model=tdModel$model)
```

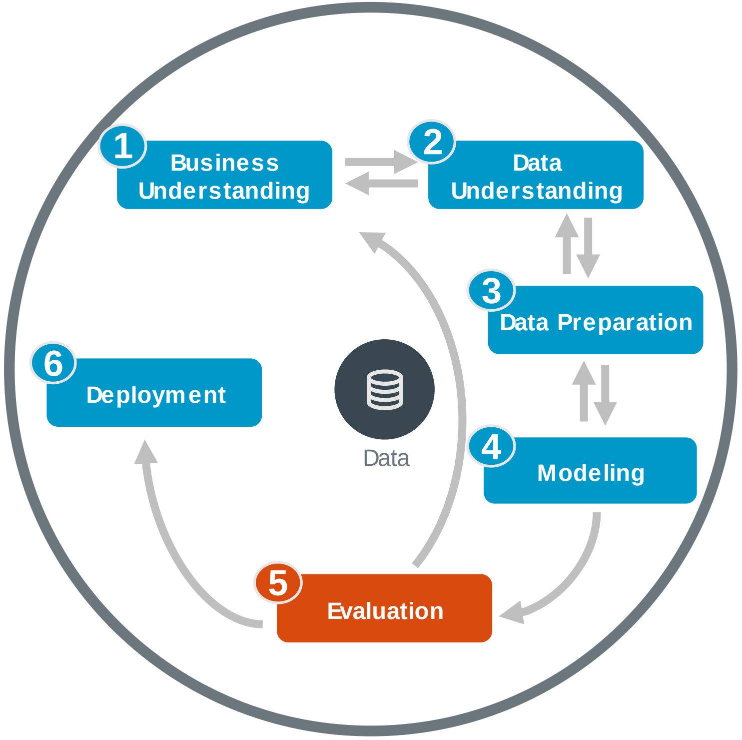
Note: Here we are validating the Accuracy of the Model. We are using math to calculate how far off each prediction is from the actual value, both from a "raw" value perspective as well as from a "percentage" perspective

Step 5. Evaluation

5. Evaluation – Assessing the business value of the model/analysis output

   a) Is Output Actionable (sufficiently meet business goals/success criteria), nice to know or new starting point?

Let's Evaluate the Output…



CRISP-DM

Step 5. Evaluation
Model Output – Is It Actionable?

teradata.

- Our model does a very poor job at meeting our stated business goal and predicting the chosen dependent variable
- Next steps could include one or more of the following:
  - Re-confirm that source data is complete and correct
  - Re-run the model with different arguments
  - Re-run the model without removing outliers
  - Re-run the model without scaling the data
  - Build other models using different predictive functions
  - Re-assess how important the stated business goal actually is

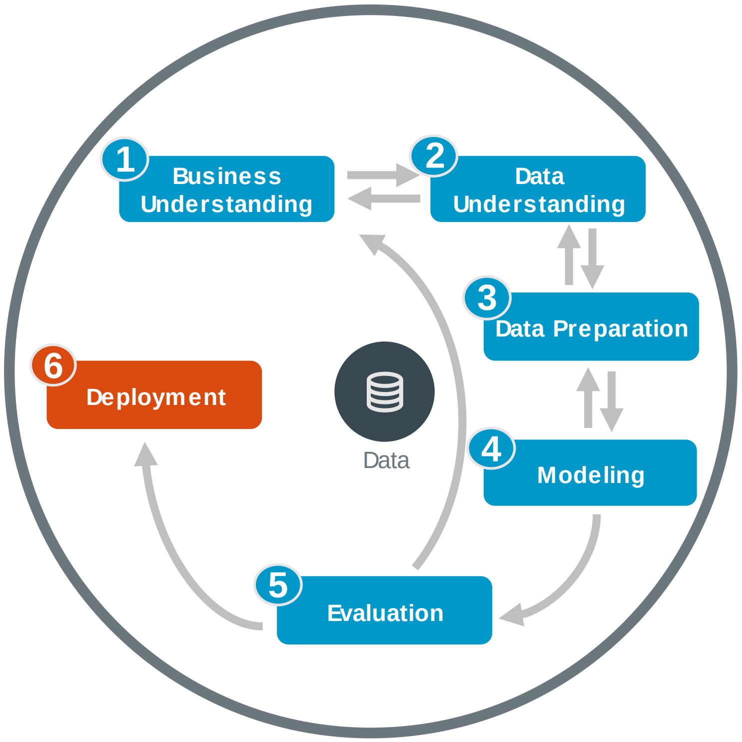
Step 6a. Deployment (Operationalizing/Monitoring/Maintenance)

teradata.

6. **Deployment** – The end goal is to "operationalize" the analytic findings. Taking analytics from insight to impact – the process of getting analytics out to business stakeholders for use/reuse to meet business goals.

   a) **Plan deployment (how to operationalize)**

      Note: This varies by customer and is not covered in this course



CRISP-DM

Step 4. Modeling
Lab 06: Scoring

teradata.


```
# Score New Dataset
tdScore <- td_lin_reg_predict_valib(data=tbl_test, model=tdModel$model,
response.column="valor_estim")
```

Current Topic – Summary and Review

- Introduction
- House Pricing Model
  - Data Science Process
    - SQL
    - Python
    - R
- **Summary and Review**

Summary

In this module, you learned how to:

- Write queries in SQL, Python, and R using the following analytic functions:
  - **Vartran**
  - **Values**
  - **DataExplorer**
  - **Matrix**
  - **Linear**
  - **LinearScore**
- Discuss the Data Science Process as applicable

Thank you.

teradata.