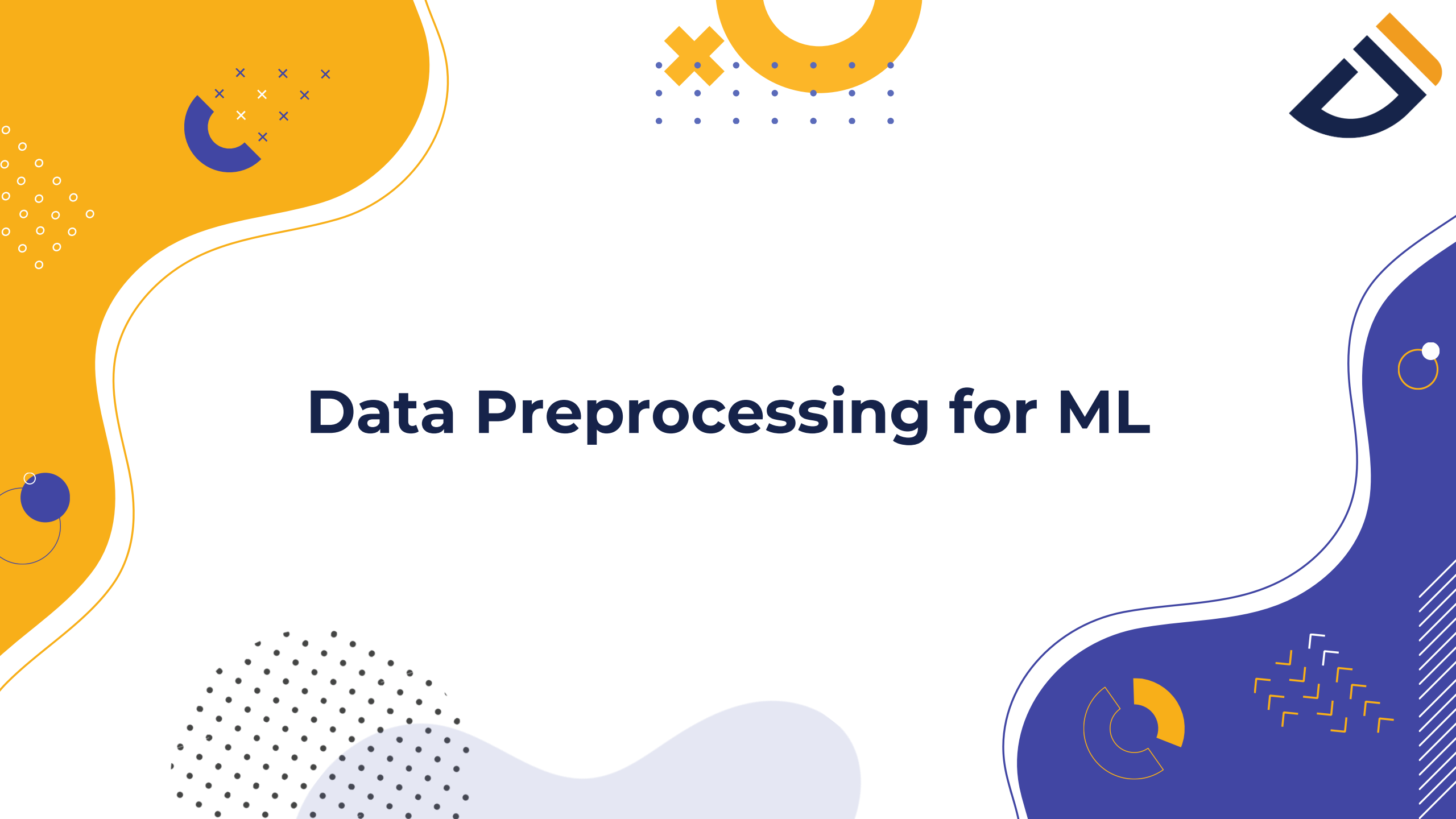


Data Preprocessing for ML





Hello!

I am Agil Haykal



I am a Data expert with extensive experience in multiple industries such as marketplace, insurance, banking, general taxation, consulting, and training.

In total, I trained more than 300 data scientists, engineers, and analysts.



Quote of the day



Garbage in garbage out.



Table of Content

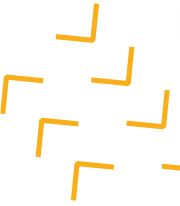
What will We Learn Today?

1. Data Preprocessing
2. Feature and Data type
3. Data Cleansing
4. One hot and Label Encoder
5. Train Test Split
6. Normalization/Standardization
7. Bias variance trade off
8. Imbalanced Data



CRISP-DM

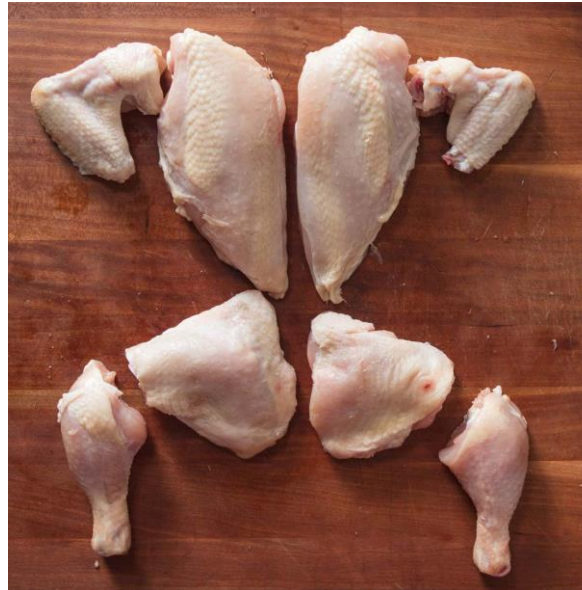




What is Data Preprocessing?



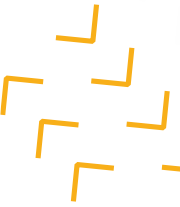
Data



Preprocessing



Machine Learning Model



What is Data Preprocessing?



Preprocessing Steps:

- Data Cleansing
- Data Transformation
- Train Test Split
- Normalization



Data Preprocessing = Feature Engineering





What is Feature?

Feature can also be called as Independent Variable or Predictor. It is basically data we use for analysis.

location	date_of_sale	property_size_sq_m	number of bedrooms	price	type
Clapham	12/4/1999	58	1	729000	apartment,1990s
Ashford	5/8/2017	119	3	699000	semi-detached,1970s
Stratford-on-Avon	29/3/2012	212	3	540000	detached,17th century
Canterbury	1/7/2009	95	2	529000	teraced,1960s
Camden	16/12/2001	54	1	616000	apartment,2000s
Rugby	1/3/2003	413	7	247000	detached, 19th century
Hampstead	5/3/2016	67	2	890000	terraced, 19th century

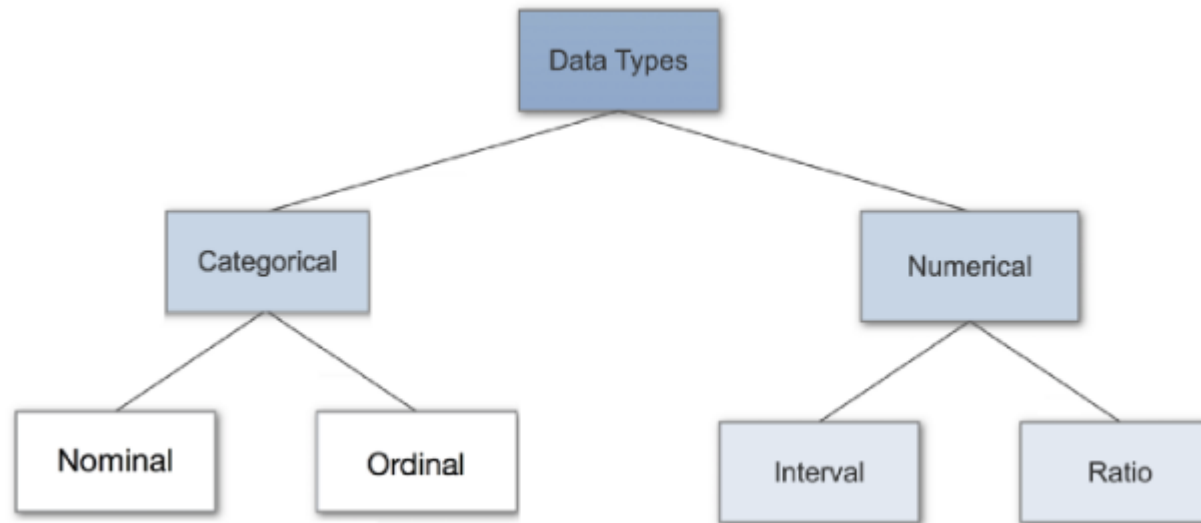


Data Type

Data type is important concept in Statistics/ML

Why understanding data type is important?

- To correctly apply statistical measurements to the data
- Therefore to correctly conclude certain assumptions about it





Data Type

Categorical VS Numeric

Categorical (Qualitative) : Consists of unordered or ordered discrete categories

Numeric (Quantitative) : Measurable in terms of numbers

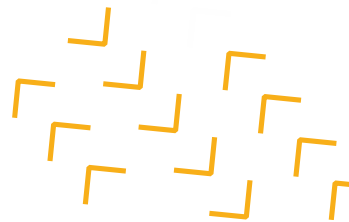
Measurement Scale of Data

Nominal : Qualitative, unordered (Gender: Male, Female)

Ordinal : Qualitative, ordered (Education: Primary, Secondary, High)

Interval : Quantitative, without absolute 0 (Temperature)

Ratio : Quantitative, with absolute 0 (Weight-kg, Height-inch)



Challenge 1: Numeric vs Categorical

location	date_of_sale	property_size_sq_m	number of bedrooms	price	type
Clapham	12/4/1999	58	1	729000	apartment,1930s
Ashford	5/8/2017	119	3	699000	semi-detached,1970s
Stratford-on-Avon	29/3/2012	212	3	540000	detached,17th century
Canterbury	1/7/2009	95	2	529000	teraced,1960s
Camden	16/12/2001	54	1	616000	apartment,2000s
Rugby	1/3/2003	413	7	247000	detached, 19th century
Hampstead	5/3/2016	67	2	890000	terraced, 19th century

Challenge 2: Measurement Scale

location	date_of_sale	property_size_sq_m	number of bedrooms	price	type
Clapham	12/4/1999	58	1	729000	apartment,1930s
Ashford	5/8/2017	119	3	699000	semi-detached,1970s
Stratford-on-Avon	29/3/2012	212	3	540000	detached,17th century
Canterbury	1/7/2009	95	2	529000	teraced,1960s
Camden	16/12/2001	54	1	616000	apartment,2000s
Rugby	1/3/2003	413	7	247000	detached, 19th century
Hampstead	5/3/2016	67	2	890000	terraced, 19th century

Feature vs Target

Target can also be called as Dependent Variable or Label. It is basically data we want to predict.

BRAND	TYPE	CYLINDER	ENG-SIZE	STROKE
Brand-A	sedan	four	109	3.4
Brand-A	sedan	five	136	3.4
Brand-B	sedan	four	108	2.8
Brand-B	sedan	four	108	2.8
Brand-C	hatchback	three	61	3.03
Brand-C	hatchback	four	90	3.11
Brand-D	hatchback	four	90	3.23
Brand-D	hatchback	four	90	3.23

Input



$f(x)$

PRICE	RISK
13950	POS
17450	POS
16430	POS
16925	POS
5151	NEG
6295	NEG
5572	NEG
6377	NEG

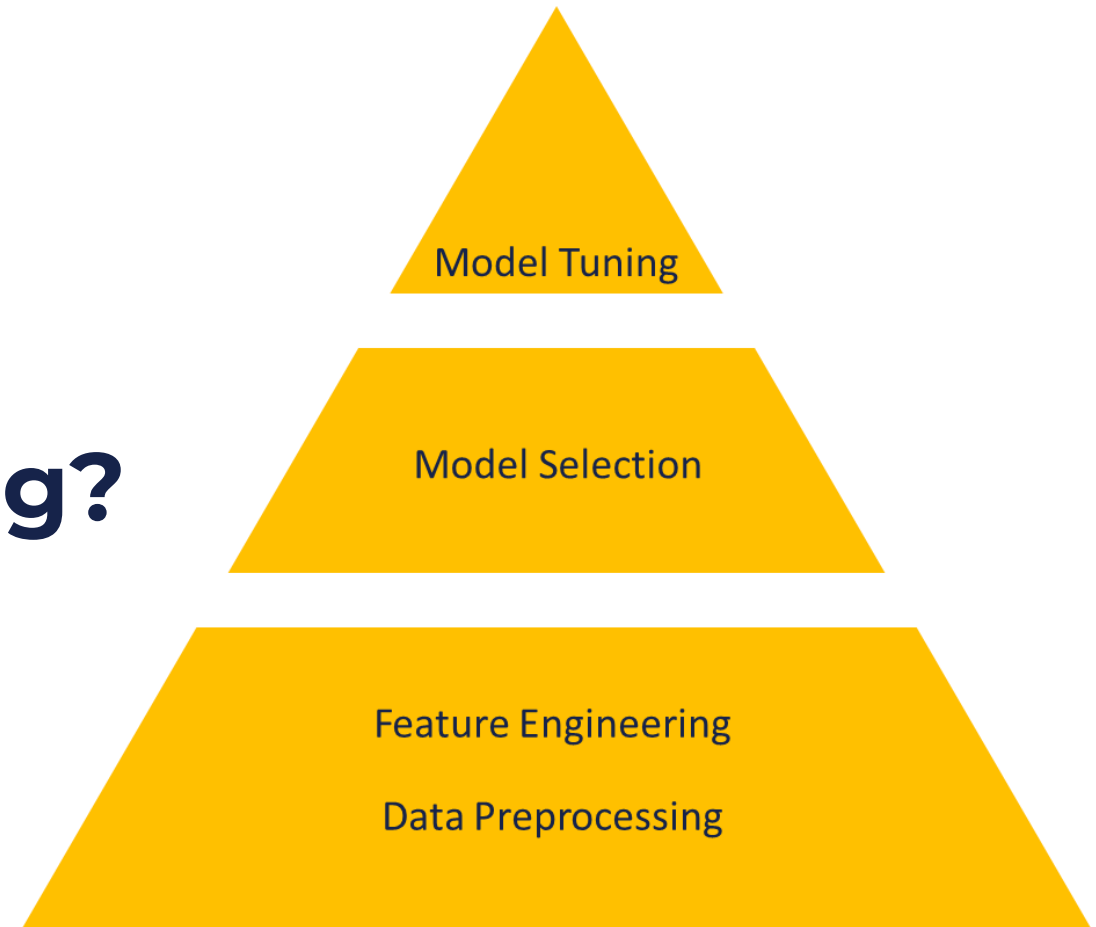
Label

$f(x)$: Function that **map** inputs to desired outputs.

Label : Price (Regression), Risk (Classification)



How important is Data Preprocessing?





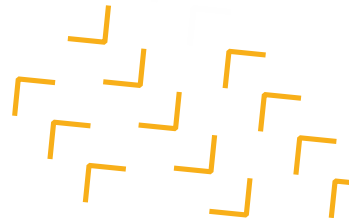
Data Cleansing

Missing Values & Data Duplication Handling

- Replace missing values with aggregated data within
- Drop duplicated data and keep the first
- Drop 'Cruise line' column, because we are going to use one categorical column

```
ship_train.loc[ship_train['Tonnage'].isnull() == True, 'Tonnage'] = ship_train['Tonnage'].mean()  
  
ship_train.drop_duplicates(keep='first')
```

```
#drop cruise_line from dataset  
ship_train=ship_train.drop(['Cruise_line'], 1)  
ship_test=ship_test.drop(['Cruise_line'], 1)
```





One Hot Encoding / Dummy

Categorical Data Transformation

Due to machine learning python only process numerical data, so categorical variable need to be converted to numerical data with process one hot encoding with function `get_dummies`. This works well for **Nominal Data Type**.

```
ship_model=pd.get_dummies(ship_train, columns=['Size'], drop_first=True)  
ship_model.head()
```

	Ship_name	Age	Tonnage	passengers	length	cabins	passenger_density	crew	Size_moderate	Size_small
0	Journey	6	30.277	694	594	355	42.64	355	0	1
1	Quest	6	30.277	694	594	355	42.64	355	0	1
2	Celebration	26	47.262	1486	722	743	31.80	670	0	1
3	Conquest	11	110.000	2974	952	1488	36.99	1910	1	0
4	Destiny	17	101.353	2642	892	1321	38.36	1000	1	0

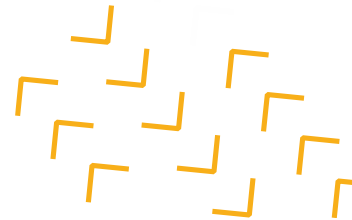


One Hot Encoding / Dummy

id	color
1	red
2	blue
3	green
4	blue



id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0





Label Encoding

Categorical Data Transformation

Basically same as One hot encoding (to change categorical data). However it is used for **Ordinal Data Type**. It change categories into numbers (1,2,3,...)

original dataset

x ₁	x ₂	y
5	8	calabar
9	3	uyo
8	6	owerri
0	5	uyo
2	3	calabar
0	8	calabar
1	8	owerri

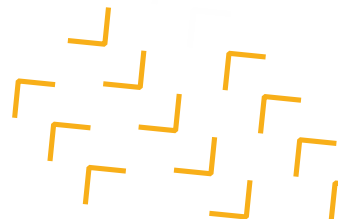
LabelEncoder



```
{  
  "calabar" ---> 0  
  "owerri" ---> 1  
  "uyo" ---> 2  
}
```

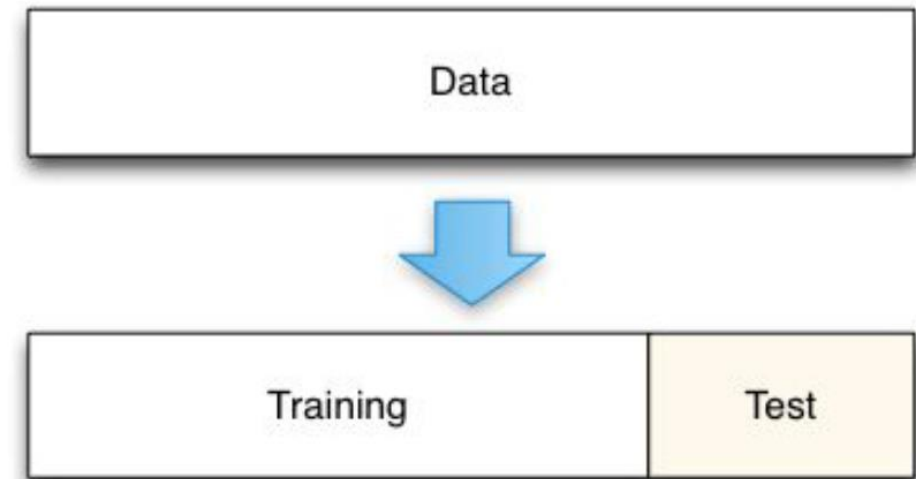
dataset with encoded labels

x ₁	x ₂	y
5	8	0
9	3	2
8	6	1
0	5	2
2	3	0
0	8	0
1	8	1



Train Test Split

- Training is a process of a machine learn the data
- The result of training is called machine learning model
- To prove accuracy of the model, data test is needed
- training set—a subset to train a model.
- test set—a subset to test the trained model.
- Due to lack of data, train-test split is necessary
- Usually proportion of train test data is 4:1 / 3:1 / 7:3





Train Test Split Analogy

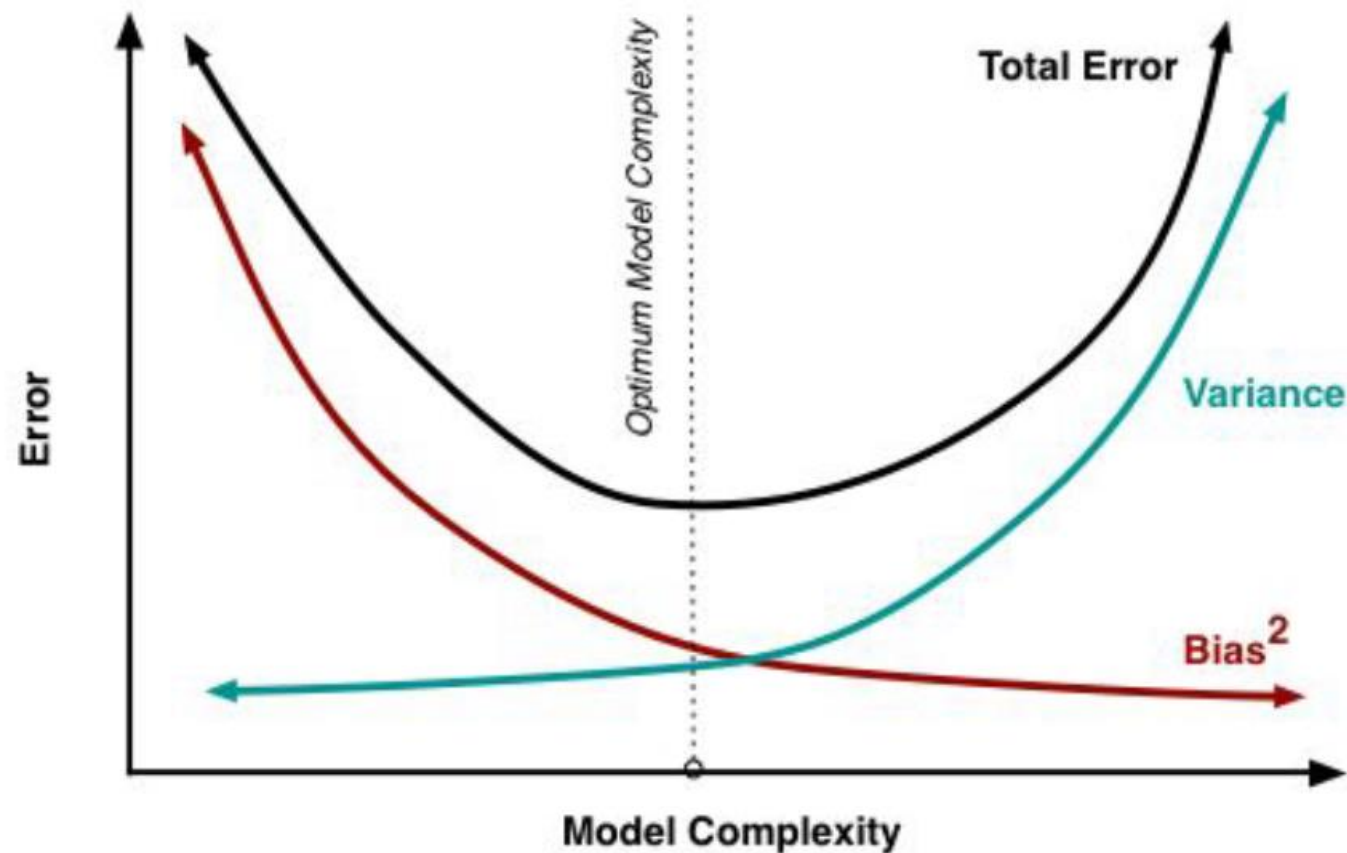


Training



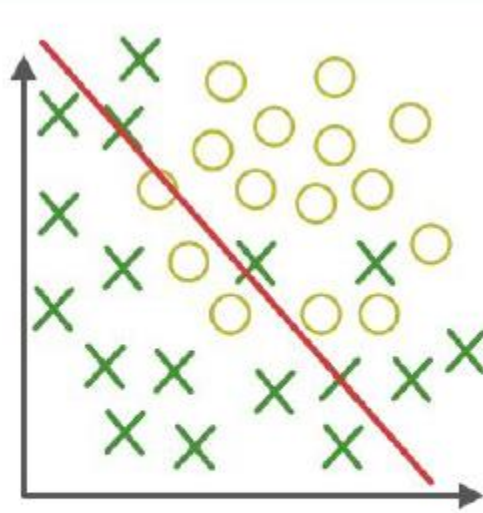
Testing / Proving

Why Train Test Split?

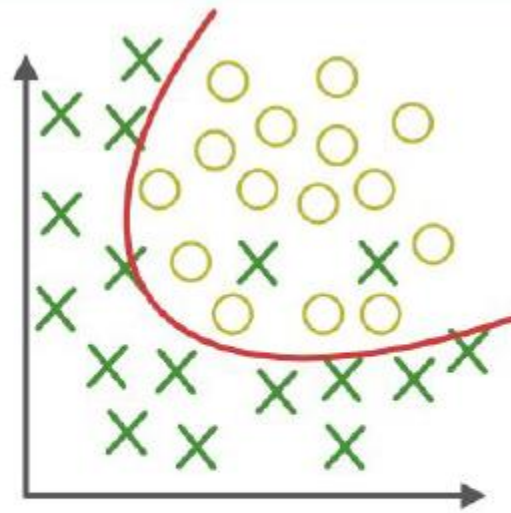




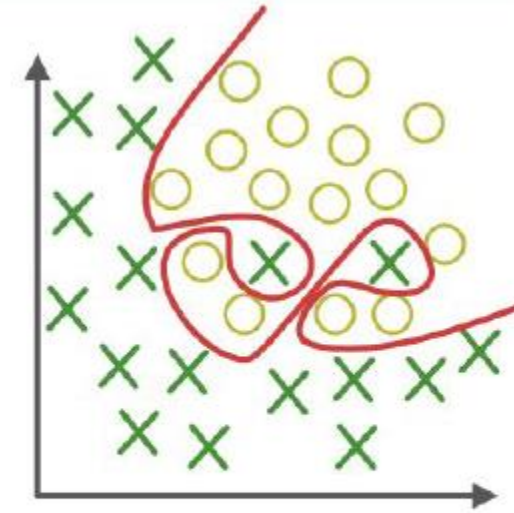
What is effect of bias variance trade off?



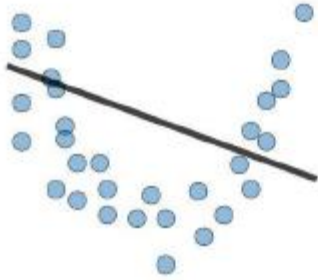


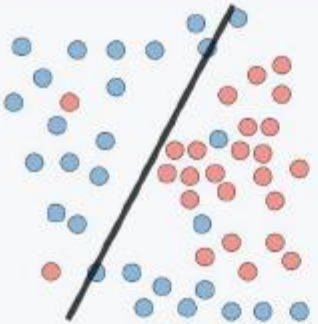
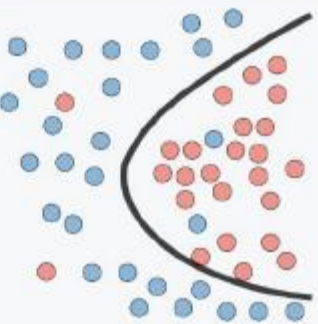
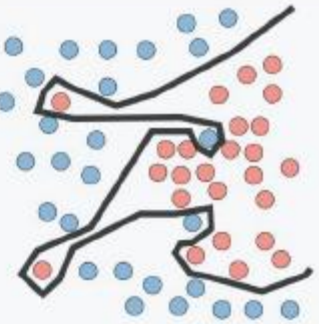

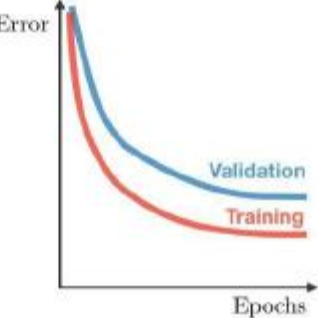
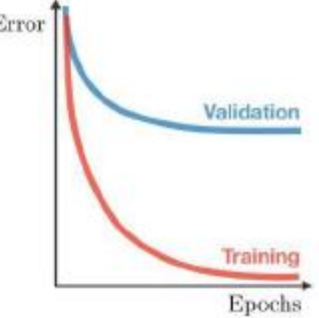
Under-fitting
(too simple to explain the variance)



Appropriate-fitting



Over-fitting
(forcefitting--too good to be true) 

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"> • Complexify model • Add more features • Train longer 		<ul style="list-style-type: none"> • Perform regularization • Get more data



Normalization/Standardization

These are used to handle distance based model or to reduce memory size of a dataset.

Normalization

Is a process to change the values of a feature into a scale (usually 0 - 1).

It considers min and max of data. Do not use this if you have outliers.

Standardization

Is a process to change the values of a feature from its mean and standard deviation.

It considers how centralized the data. Outliers can be handled in here.





Normalization/Standardization

- **Min-Max Scaling**

Uses MinMaxScaler

Transform to defined range

$$y = \frac{x - \min x_i}{\max x_i - \min x_i}$$

Where

\bar{x} = mean

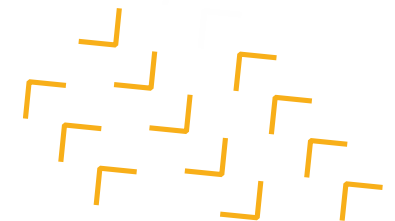
s = Standard deviation

- **Standardization**

Uses StandardScaler

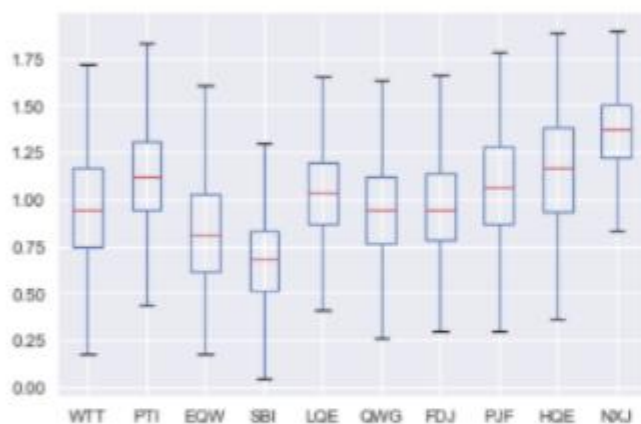
Transform to mean=0, sd=0

$$y = \frac{x - \bar{x}}{s}$$



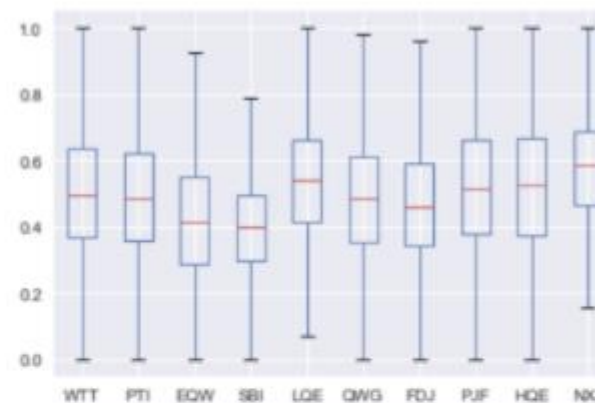


Normalization/Standardization

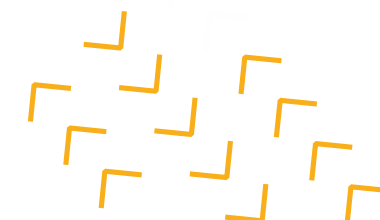
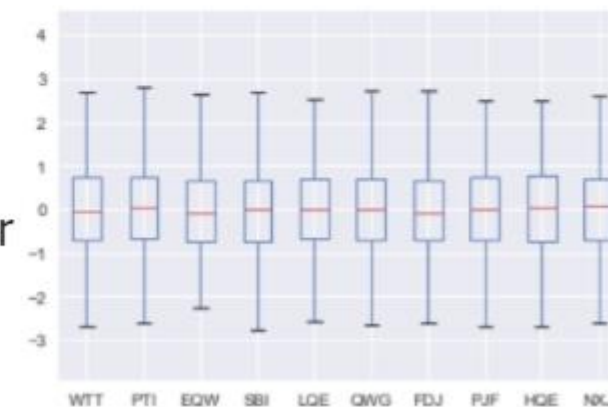


Original Data

MinMaxScaler



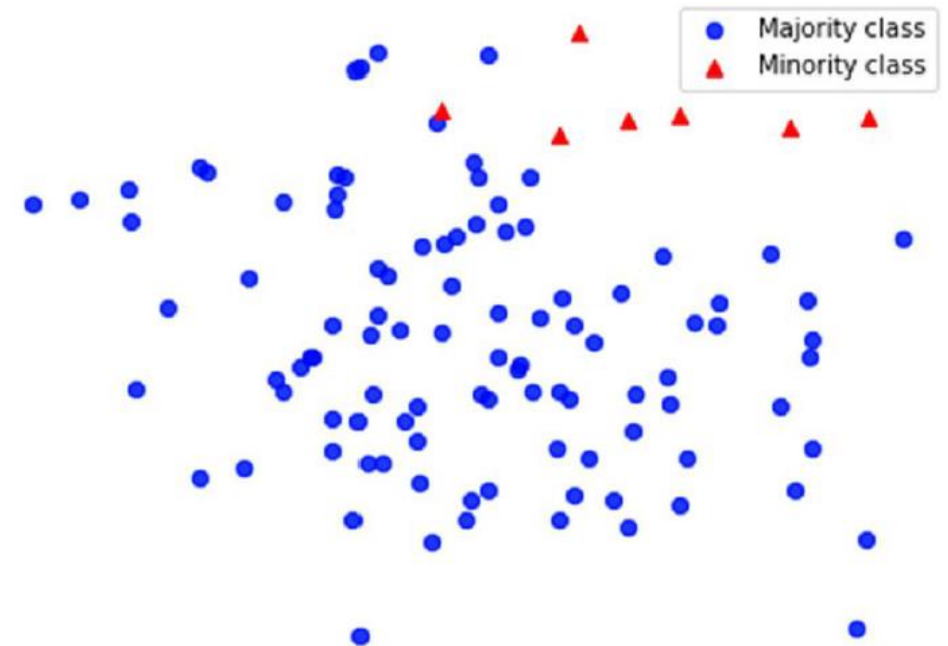
StandardScaler





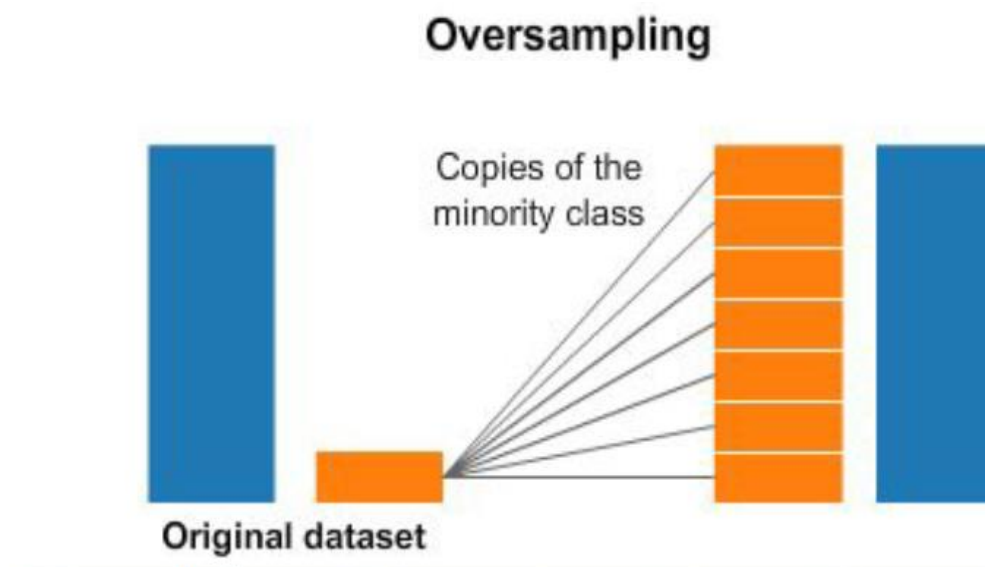
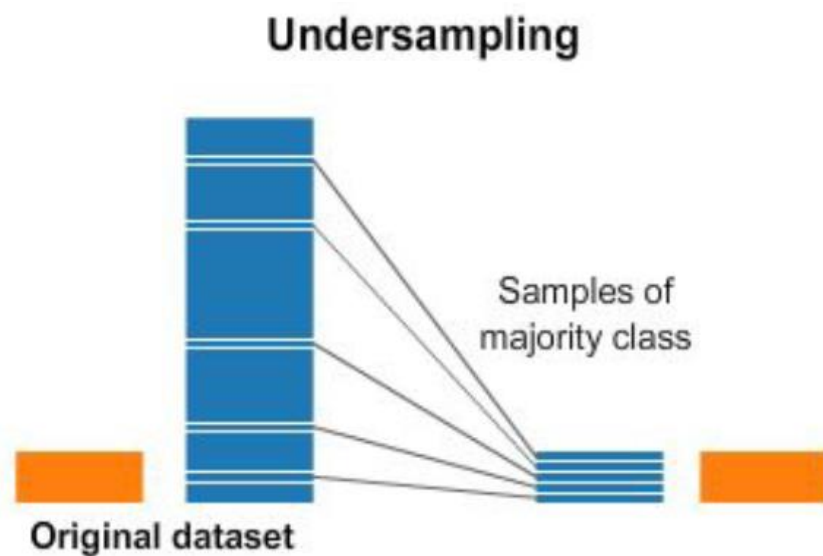
Imbalanced Dataset

- It is a common case of Classification model where proportion of each class in a target is imbalanced.
- This kind of data need to be balanced.
- It must be handled only at **Training Dataset**.





Imbalanced Handling



**Thank
YOU**

