Model Building Phase

Impact of an Incident Prediction Project

- By Ashish Gore

Project Approach

01 **Data Cleaning Phase**
Feature Selection, Conversion Into Dummies

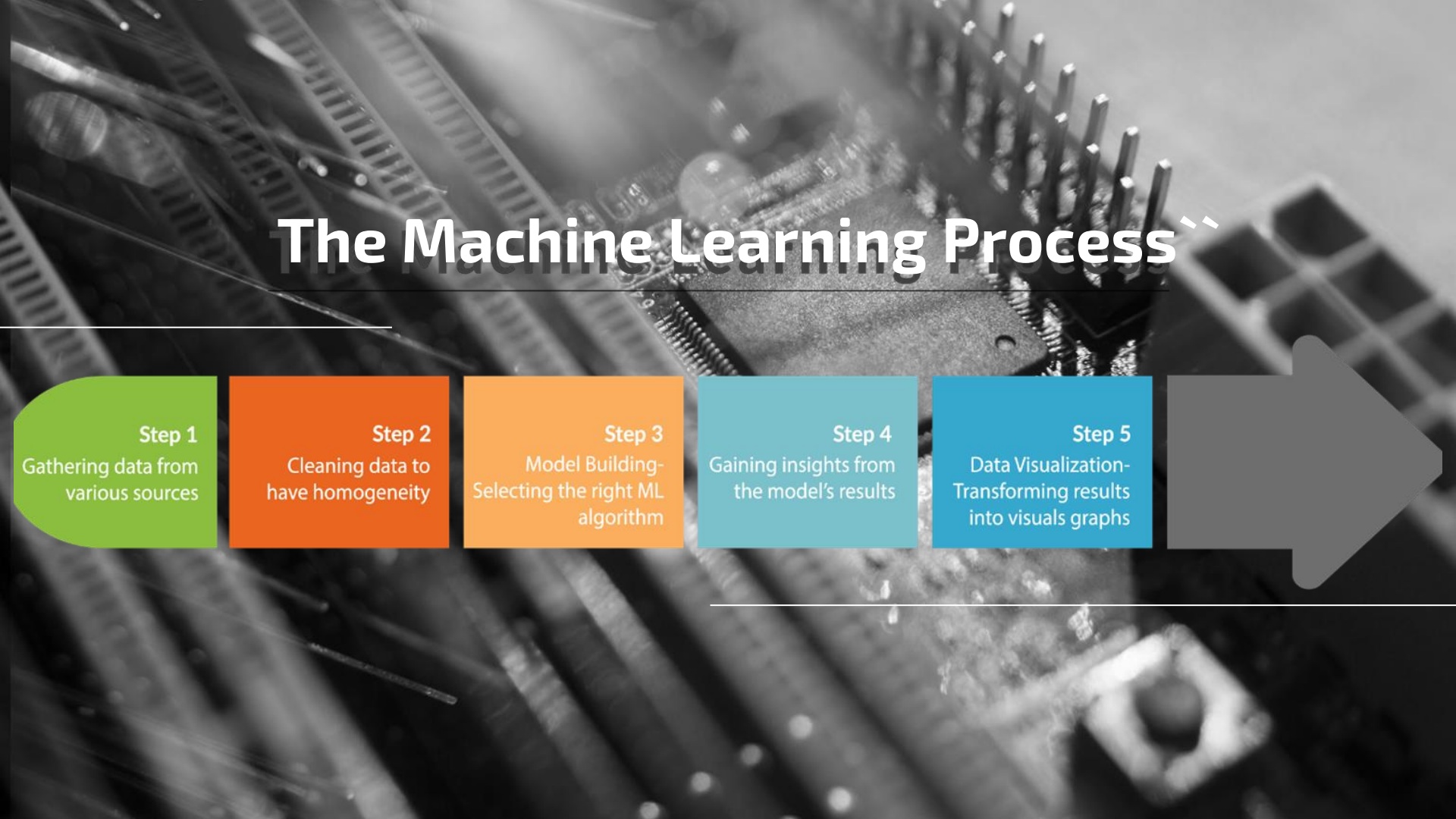
02 **Dimensionality Reduction**
Which Method & Why?

03 **Train & Test Splitting**
Sampling Methods for Imbalanced Data

04 **Algorithms**
Multiple Algorithms Used & Their Accuracies

05 **Meta Algorithms & Neural Networks**
Multiple Meta Algorithms & NN Used & Their Accuracies

06 **Cross Validation & Hyperparameter Tuning**
Which Method & Why?

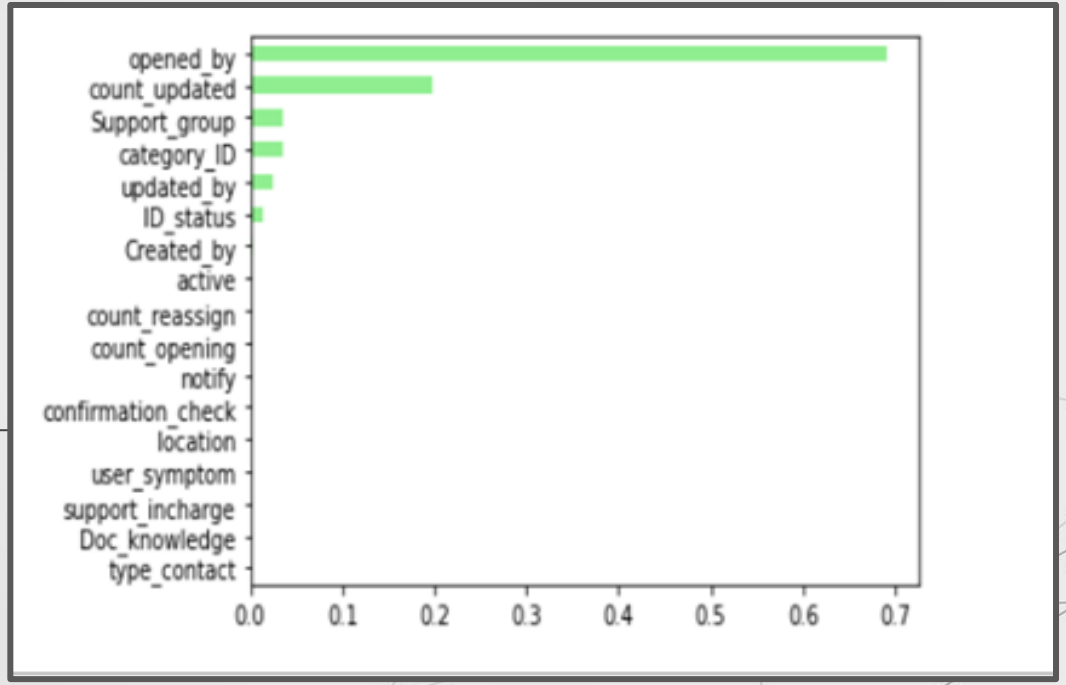

The Machine Learning Process``
Step 1
Gathering data from various sources
Step 2
Cleaning data to have homogeneity
Step 3
Model Building- Selecting the right ML algorithm
Step 4
Gaining insights from the model's results
Step 5
Data Visualization- Transforming results into visuals graphs

01

Data Cleaning Phase

Feature Selection

- Removed -100 Class from “ID_Status” Feature
- Selected only Important Features



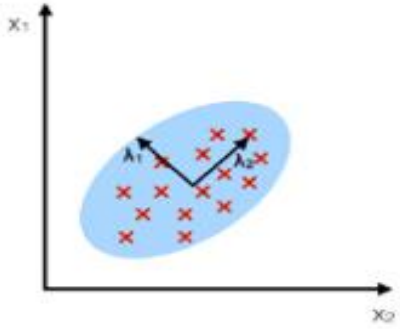
Conversion Into Dummies

- Each Class Converted into Separate Dummies
- Resulting Dimensions (141707, 534)

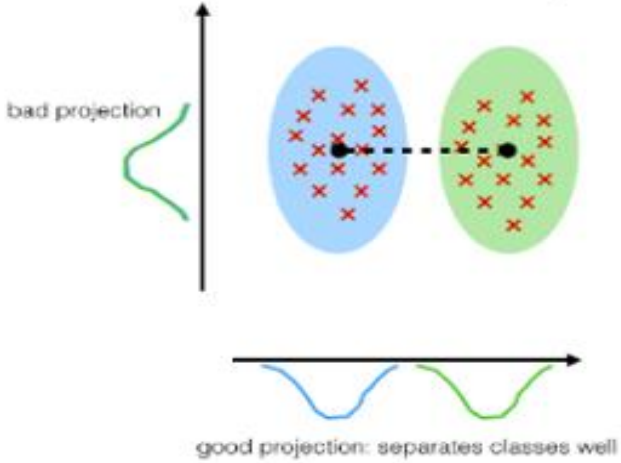
02

Dimensionality Reduction

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



LDA vs PCA

Dimensions Reduced:
From

To

53(=>

Difference between two techniques

LDA(Linear Discriminant Analysis)	PCA(Principal Component Analysis)
1. LDA is a *supervised machine learning method* that is used to separate two groups/classes.	1. PCA is an *unsupervised machine learning method* that is used for dimensionality reduction.
2. LDA attempts to find a feature subspace that maximizes class separability (variance)	2. PCA as a technique that finds the directions of maximal variance:
3. LDA looks at what type of point/ features subspace gives more discrimination to separate the data.	3. Variables are transformed into a new set of variables which is a combination of attributes from our original dataset in a way that maximum variation is retained. It is not a feature selection technique.

Which Technique did we Choose?
Why?

We Choose Both PCA & LDA Technique

WHY should you we PCA?

1. Do you want to reduce the no.of variables, but are not able to identify variables to completely remove from consideration?
2. Do you want to ensure your variables are independent of one another?
3. Are you comfortable making your independent variable less interpretable?

WHY should you use LDA?

1. LDA is a type of Linear combination, a mathematical process using various data items and applying a function to that site to separately analyze multiple classes of objects or items.
2. Fisher's Linear discriminant, linear discriminant analysis can be useful in areas like image recognition and predictive analysis in marketing.
3. LDA helps in preventative data for more than two classes, when Logistics Regression is not sufficient. The linear Discriminant analysis takes the mean value for each class and considers variants to make predictions assuming a Gaussian distribution.

03

Train and Test Splitting

Sampling and Scaling Methods

Scaling Techniques

Feature **Scaling** is a technique to standardize the independent features present in the **data** in a fixed range. It is performed during the **data** pre-processing to handle highly varying magnitudes or values or units.

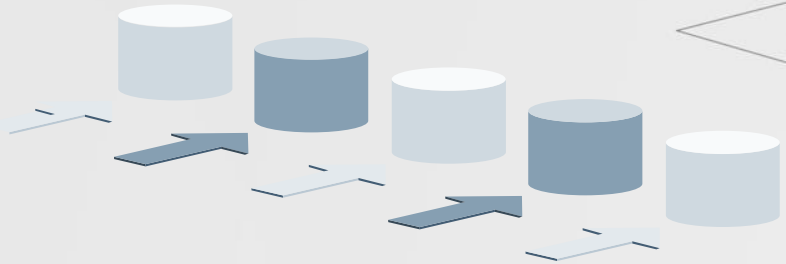
1) **Min Max Scaler**
2) **Standard Scaler**
3) **Max Abs Scaler**
4) **Robust Scaler**
5) **Quantile Transformer Scaler**
6) **Power Transformer Scaler**
7) **Unit Vector Scalar**

Difference between Minmax and Standard

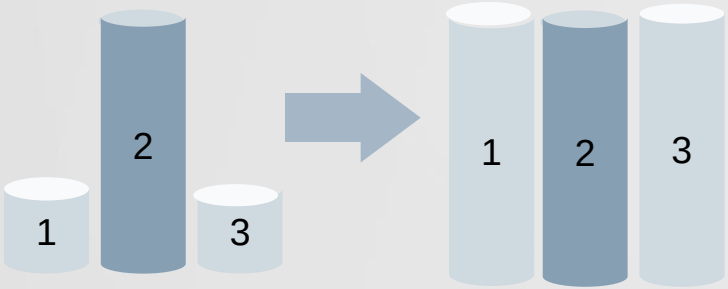
Minmaxscaler	Standardscaler
Minmaxscaler shrinks the data within the range of -1 to 1(if there are negative values)	Standard assumes that data has normally distributed features and will scale them to zero mean and 1 standard deviation.
This is used when distribution is not Gaussian, responds well if standard deviation is small.	After applying Standard Scalar all features will be of same scale .
This scaler is sensitive to outliers.	The outliers have an influence when computing the empirical mean and standard deviation which shrink the range of the feature values.

Sampling Techniques Available

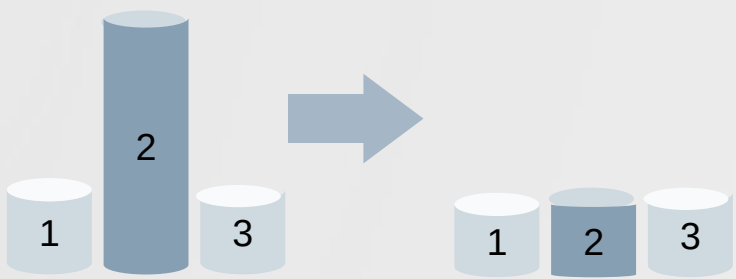
Simple Random Sampling



Over Sampling



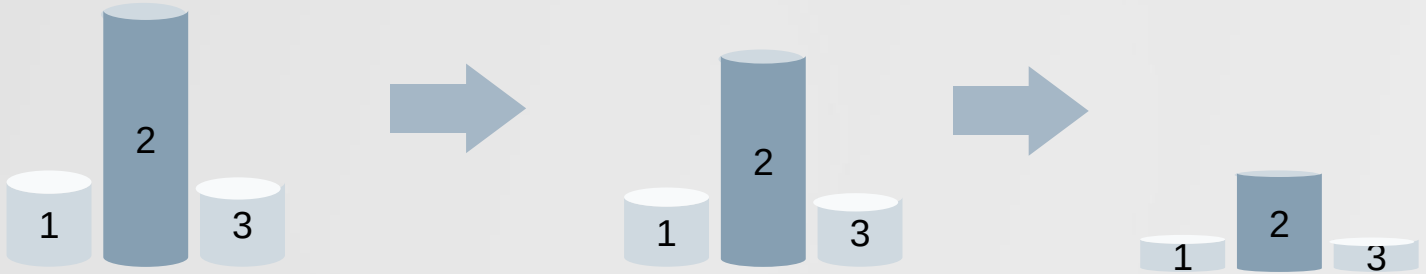
Under Sampling



Sampling Technique We Choose

Stratified Random Sampling

1000 Records



1000 Records Original

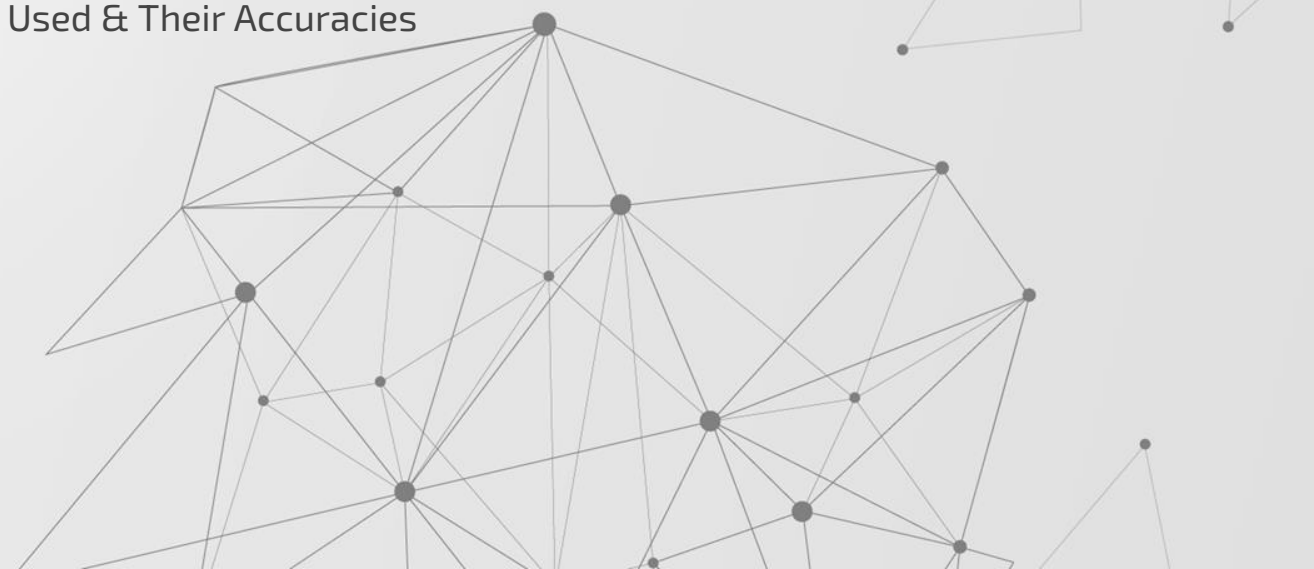
700 Records Train

300 Records Test

04

Different ML Algorithms

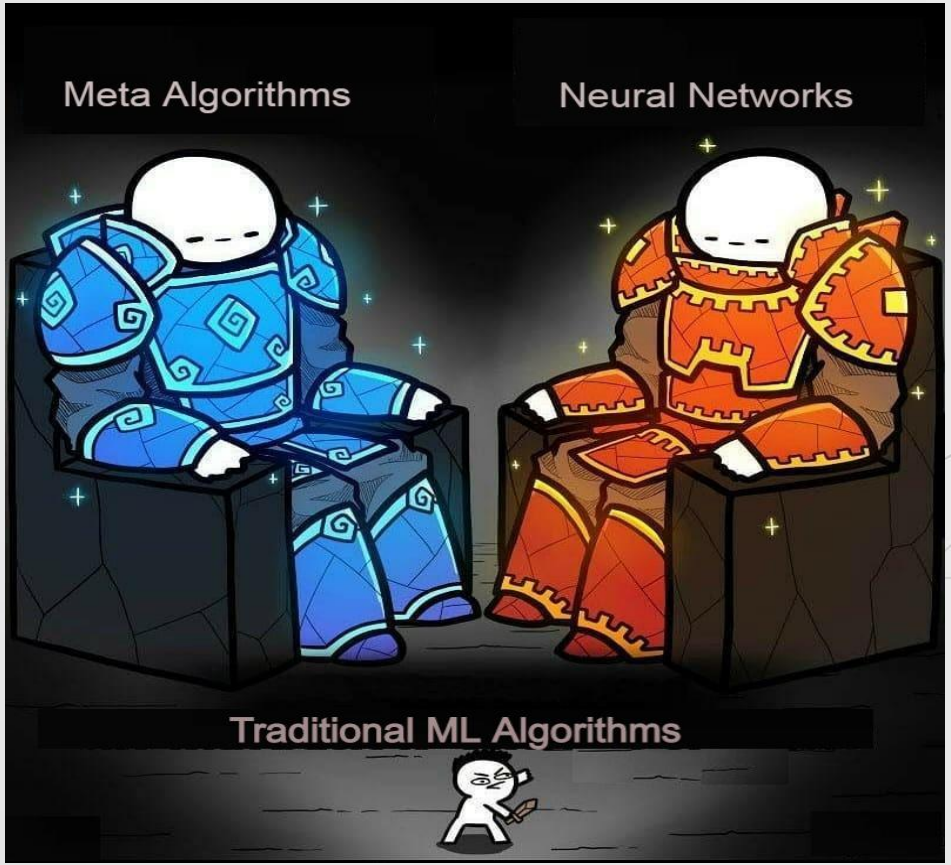
Multiple Algorithms Used & Their Accuracies



Accuracies obtained using diff Algorithms

LDA	PCA
1. Decision Tree: Accuracy 0.94	**1. Decision Tree** Accuracy 0.94
2. Random Forest Accuracy 0.95	**2. Random Forest:** Accuracy 0.95
3. Support Vector Classifier Accuracy 0.95	**3. Support Vector Classifier** Accuracy 0.95
4. Gaussian NB Accuracy 0.92	**4. Gaussian NB** Accuracy 0.90
5. KNN Accuracy 0.95	**5. KNN** Accuracy 0.95

Traditional vs Advanced ML Algorithms



05

Meta Algorithms & Neural Networks

Multiple Meta Algorithms & NN used along with their Accuracies

Accuracies obtained using diff Meta-Algorithms

LDA	PCA
1. Voting Classifier Accuracy of Voting Classifier: 0.947	1.**Voting Classifier** Accuracy of Voting Classifier: 0.954
2. Bagging Classifier Accuracy of Bagging Classifier: 0.949	**2. Bagging Classifier** Accuracy of Bagging Classifier: 0.954
3. AdaBoost Classifier Accuracy of Bagging Classifier: 0.949	**3. AdaBoost Classifier** Accuracy of Bagging Classifier: 0.954
4. XGBoost Classifier Accuracy of Bagging Classifier: 0.953	**4. XGBoost Classifier** Accuracy of Bagging Classifier: 0.951

Neural Networks Model Building

NN Model 1

303 Params

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 10)                160
_________________________________________________________________
dense_1 (Dense)              (None, 10)                110
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 33
=================================================================
Total params: 303
Trainable params: 303
Non-trainable params: 0
_________________________________________________________________
```

NN Model 2

22,103 Params

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 100)               1600
_________________________________________________________________
dense_4 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_5 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 303
=================================================================
Total params: 22,103
Trainable params: 22,103
Non-trainable params: 0
_________________________________________________________________
```

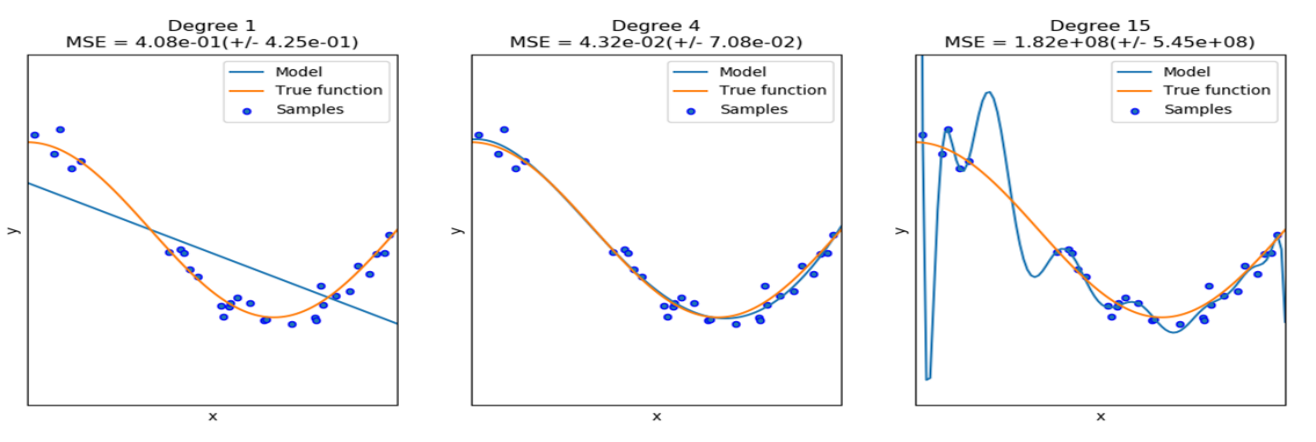
06

Cross Validation & Hyperparameter Tuning

Cross Validation

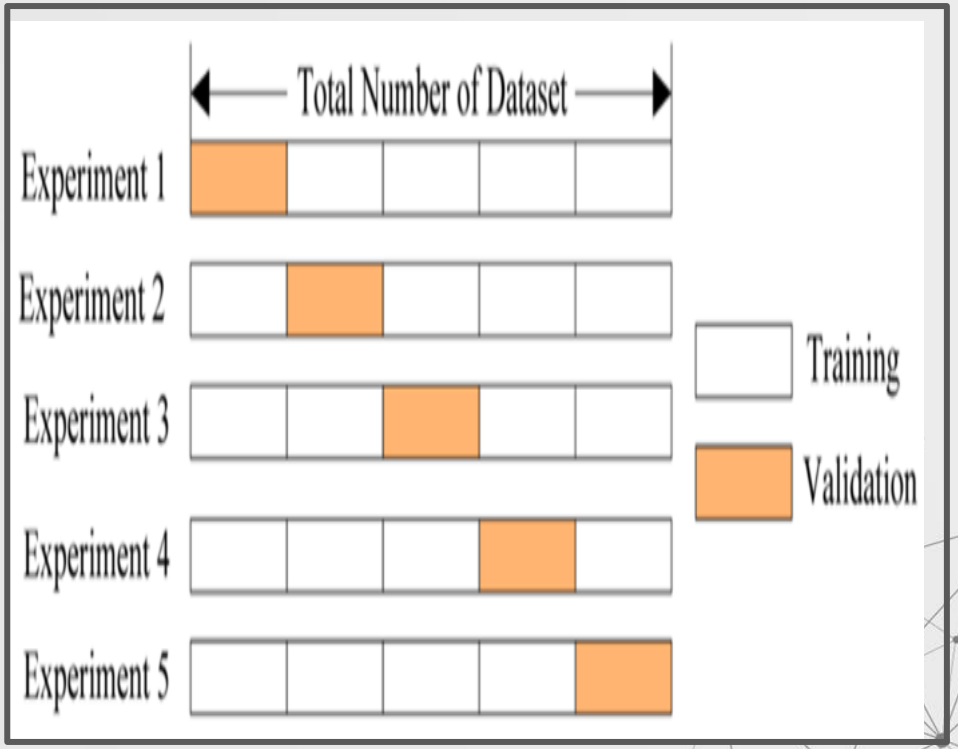
Cross Validation is a technique used to identify how well our model performed and there is always a need to test the accuracy of our model to verify that, our model is well trained with data without any overfitting and underfitting.

In statistics, **overfitting** means our model fits too closely to our data. The fitted line will go exactly through every point in the graph. **Underfitting** means our model doesn't fit well with the data(i.e, model cannot capture the underlying trend of data, which destroys the model accuracy)and occurs when a statistical model or machine learning algorithm cannot adequately capture the underlying structure of the data.



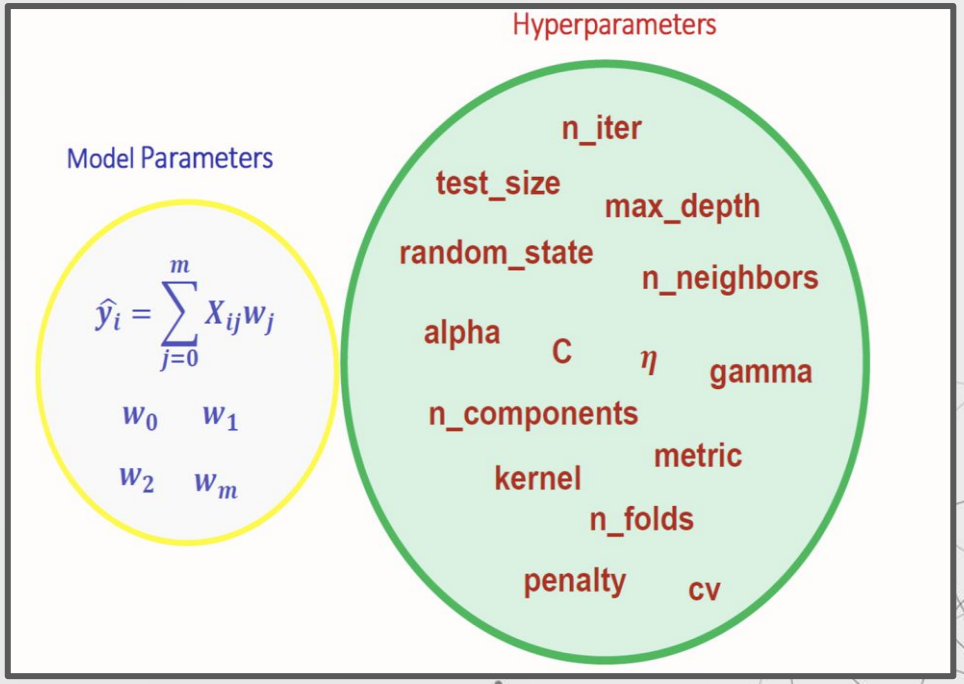
K-Fold Cross Validation

- In **K Fold cross validation**, the data is divided into k subsets and train our model on k-1 subsets and hold the last one for test.
- This process is repeated k times, such that each time, one of the k subsets is used as the test set/ validation set and the other k-1 subsets are put together to form a training set.
- We then average the model against each of the folds and then finalize our model. After that we test it against the test set
- The more the number of folds, less is value of error due to bias but increasing the error due to variance will increase; the more folds you have, the longer it would take to compute it and you would need more memory.



Hyperparameter Tuning

- Hyperparameters are external to our model and cannot be directly learned from the regular training process.
- These parameters express “higher-level” properties of the model such as its complexity or how fast it should learn.
- Hyperparameters are model-specific properties that are ‘fixed’ before you even train and test your model on data.
- Currently involves either *random search* or *grid search* across sets of hyperparameters.



We have imported two libraries ***RandomisedSearchCV*** and ***GridsearchCV*** from ***sklearn.model_selection*** *for performing k-fold cross* validation and find the best hyperparameters that describe the model from the infinite space and grid.
The hyperparameters used in the model are:

- ***RandomisedSearchCV***

1. Best parameters found: ***n_estimators***(no of trees used in the model) -100
   max_depth(depth or layers of trees used) - 7
   learning_rate(how quickly the model is adapted to the problem.): 0.05.
2. Best Score found: **0.9770745552016705**.

- ***GridsearchCV***

1. Best parameters found: ***n_estimators*** - 100
   max_depth- 8
   learning_rate - 0.05
1. Best Score found: **0.9771519348716857**.

A Quick Tour to Codes

Thank You!!