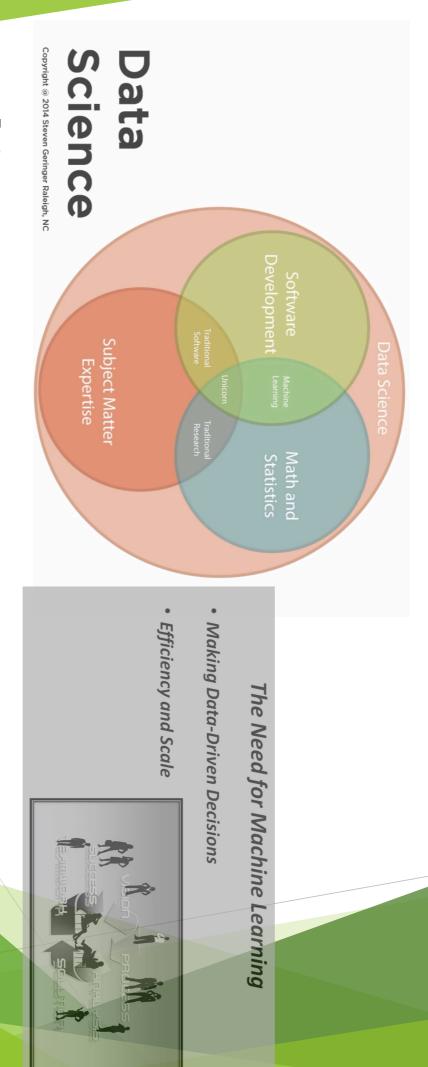


AIML - Data pre-processing and building a prediction model using AIML

Murali Krishna

Basics of Machine Learning



Facts :-

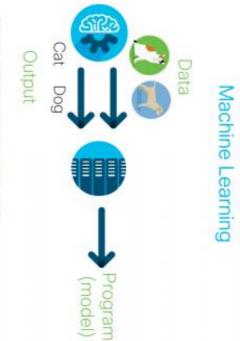
1952: Arthur Samuel, who was the pioneer of machine learning, created a program that helped an IBM computer to play a checkers game. It performed better more it played. $_{
m 2}$

1959: In 1959, the term "Machine Learning" was first coined by Arthur Samuel

Basics of Machine Learning

AI/ML Writes Code Based on Examples



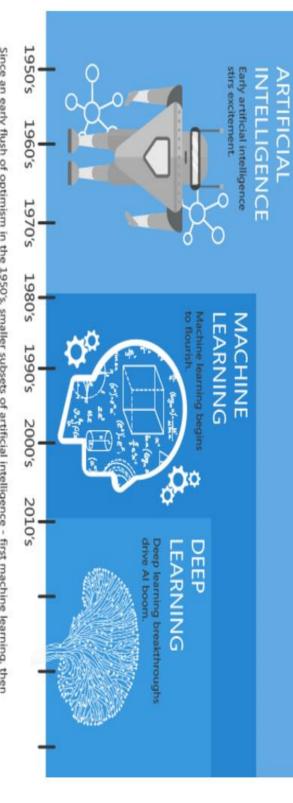


Machine Learning generates code:
Based on lots of examples with correct answers

"Cats have tail, pointy ears, tail and meow"
"Dogs have tail, pointy snout, and woof"

Programmer describes in code:

What is Machine learning



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

What is Machine Learning?

Learn From Experience



Follow Instructions

Data Learn From Experience



Machine learning applications



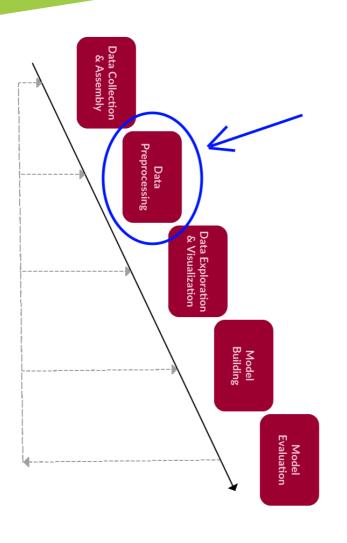


Machine learning workflow

Ask the right question Prepare the data Select the algorithm Train the model Test the model

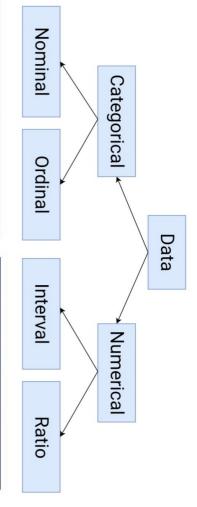
Data Pre-Processing

data gets transformed, or Encoded, to bring it to such a state that now the easily interpreted by the algorithm. In any Machine Learning process, Data Pre-processing is that step in which the machine can easily parse it. In other words, the features of the data can now be



- Importing the dataset
- Handling missing data
- Handling categorical data
- Splitting the dataset into training and test dataset
- Feature Scaling

Kinds of data



Categorical variables without any implied order without any implied order implied order Example : A new car model comes in these colors : Black, Blue, White, Silver Medium - Small Categorical variables with a natural implied order but the scale of difference is not defined Example : Sizes of clothes has a natural order : Extra Small < Small < Small < Extra Large But this does not mean Large - Medium - Small			
Catego variables natural in order bu scale difference define Examp Sizes of chas a na order: B Small < S Medium < Extra L But this not mean - Medium -	Example : A new car model comes in these colors : Black, Blue, White, Silver	Categorical variables without any implied order	Nominal
with a with a mplied at the of sis not ed of stural extra stural extra small < Large arge - Large - Small = Small	Example: Sizes of clothes has a natural order: Extra Small < Small < Medium < Large < Extra Large - But this does not mean Large - Medium = Medium = Small	Categorical variables with a natural implied order but the scale of difference is not	Ordinal

7	
Examples : Calender Dates, Temperature in Celsius or Farhenheit	Numeric Numeric variabes with a definied unit of measurement, so the differences between values are meaningful
Examples: Temperature in Kelvin, Monetary quantities, Counts, Age, Mass, Length, Electrical Current	Numeric Numeric variables with a defined unit of measurement but both differences and ratios are meaningful



Importing libraries

```
# used for splitting training and testing data
from sklearn.model_selection import train_test_split
                                                                                                                                                                                                                                                                                                                                                                                                                                                    import pandas as pd
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               import numpy as np
                                                                                                                                                                                                                                                                                                                                                                             # used for handling missing data
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           # used for handling the dataset
                                                                                                                                                                                                                                                               # used for encoding categorical data
                                                                                                                                                                                                                                                                                                                                      from sklearn.impute import SimpleImputer
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        # used for handling numbers
from sklearn.preprocessing import StandardScaler
                                  # used for feature scaling
                                                                                                                                                                                                                        from sklearn.preprocessing import LabelEncoder, OneHotEncoder
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Importing libraries
```

Importing data

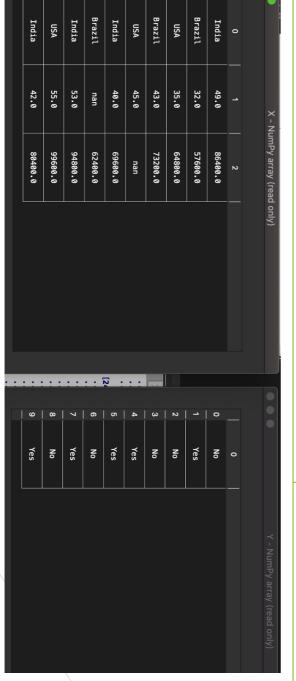
```
dataset = pd.read_csv('Data.csv')
                                       # to import the dataset into a variable
```

Splitting the attributes into independent and dependent attributes

attributes to determine dependent variable / Class
X = dataset.iloc[:, :-1].values

dependent variable / Class
Y = dataset.iloc[:, -1].values

11	10	9	∞	7	6	5	4	ω	2	1
11 India	USA	India	Brazil	India	USA	Brazil	USA	Brazil	India	Region
42	55	53		40	45	43	35	32	49	Age
80400 Yes	99600 No	94800 Yes	62400 No	69600 Yes		73200 No	64800 No	57600 Yes	86400 No	Income
Yes	No	Yes	No	Yes	Yes	No	No	Yes	No	Online Shopper



Handling missing data

handling the missing data and replace missing values with nan from numpy and replace with

mean of all the other values

imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer = imputer.fit(X[:, 1:])

X[:, 1:] = imputer.transform(X[:, 1:])



Note :- Missing Data can be replaced by Statistics – Mean, Median, Mode, IQR

Handling categorical data

Encode categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

Label encoding
labelencoder_X = LabelEncoder()

X[:, 0] = labelencoder_X.fit_transform(X[:, 0])

One hot encoding

onehotencoder = OneHotEncoder(categorical_features=[0])

 $X = onehotencoder.fit_transform(X).toarray()labelencoder_Y = LabelEncoder()$

Y = labelencoder_Y.fit_transform(Y)

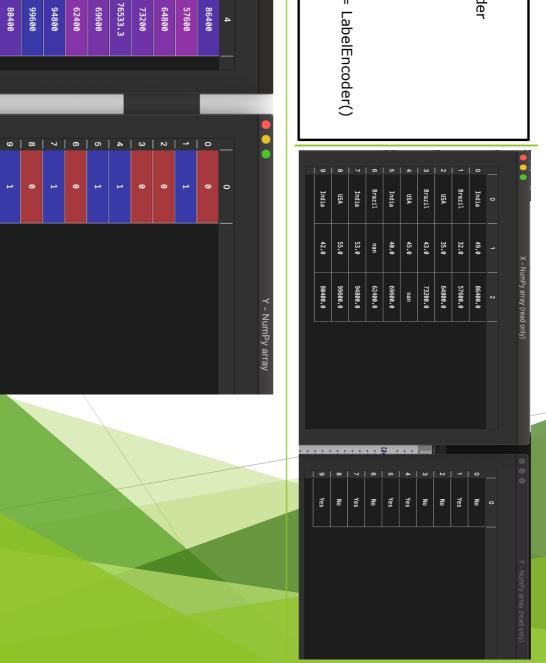
- | 0

| C | C | 4

35

၈ | **೮** |

43.7778 40

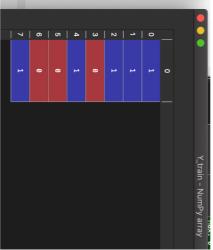


plitting the dataset into training & test

the correctness of the algorithm by testing on testing set. - we use the training set to make the algorithm learn the behaviours present in the data and check

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0) # splitting the dataset into training set and test set in the 80:20 ratio









Feature Scaling(Variable transformation)

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.

Two columns age and income that contains numerical values. We notice that the variables are not on the same scale because the age are going from 32 to 55 and the salaries going from 57.6 K to like 99.6 K.

So because this age variable in the salary variable don't have the same scale. This will cause some issues in your models. And why is that. It's because lot of machine are based on what is called the Euclidean distance.

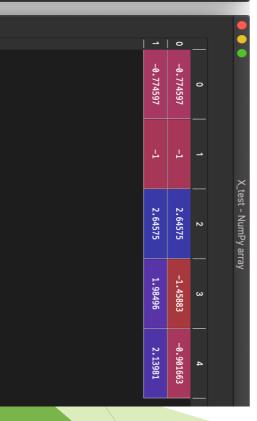
We use feature scaling to convert different scales to a standard scale to make it easier for Machine Learning algorithms.

feature scaling

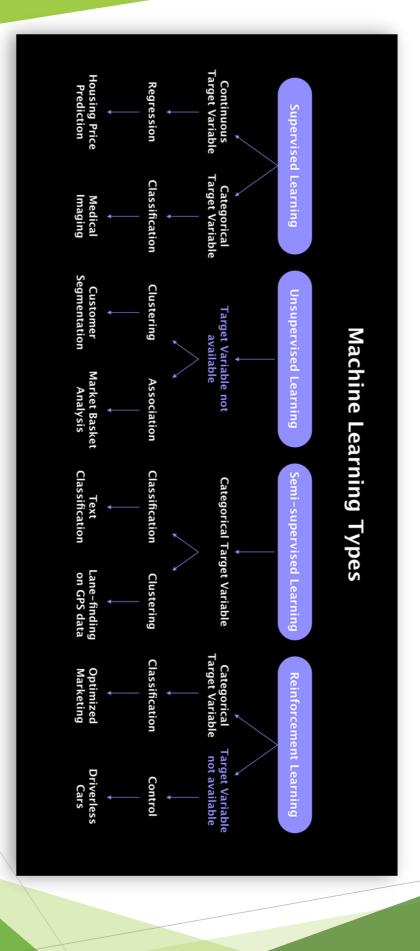
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)

 $X_{\text{test}} = \text{sc}_{X}.\text{transform}(X_{\text{test}})$

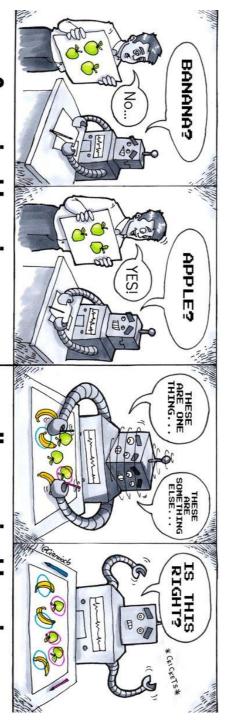




Types of Machine Learning



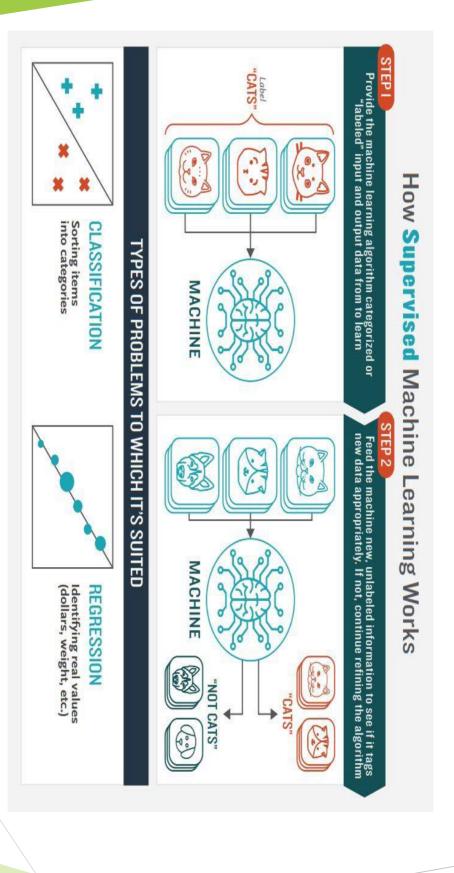
Types of Machine Learning



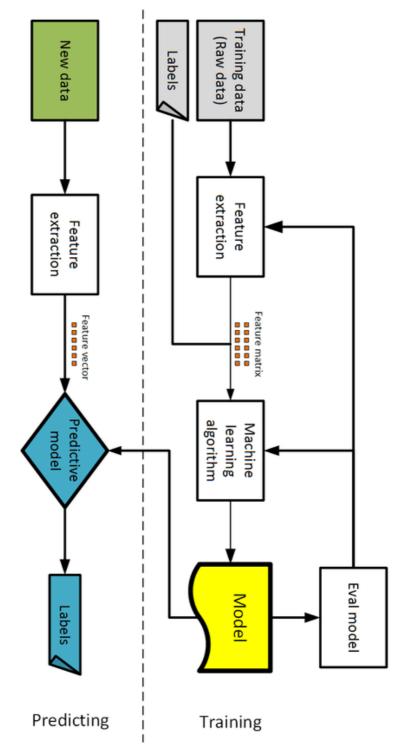
Supervised Learning

Unsupervised Learning

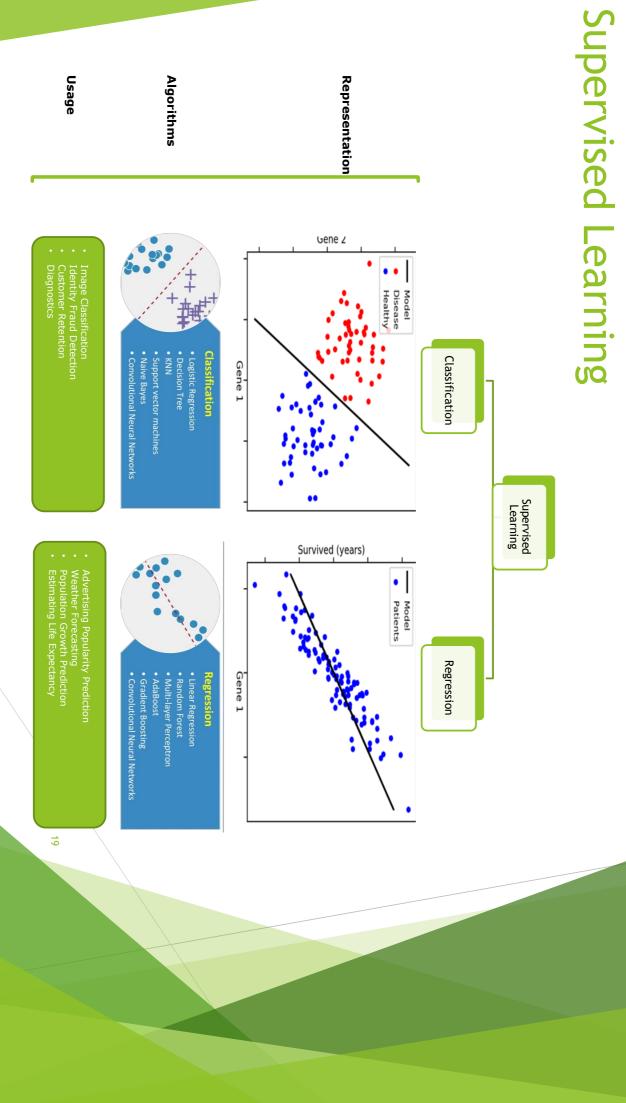
Supervised Learning



Supervised Learning - Flow chart







Unsupervised Learning

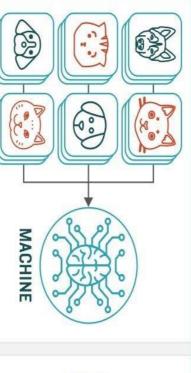


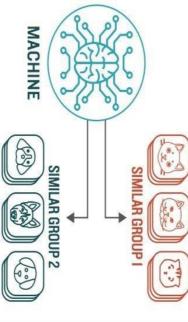


Provide the machine learning algorithm uncategorized, unlabeled input data to see what patterns it finds

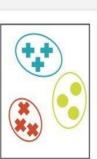
STEP 2

Observe and learn from the patterns the machine identifies





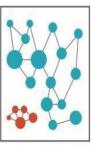
TYPES OF PROBLEMS TO WHICH IT'S SUITED



CLUSTERING

Identifying similarities in groups

For Example: Are there patterns in the data to indicate certain patients will respond better to this treatment than others?



ANOMALY DETECTION

Identifying abnormalities in data

For Example: Is a hacker intruding in our network?

Unsupervised Learning

Unsupervised Learning

Clustering

Dimensionality Reduction

K Means

Clustering

mutually exclusive spherical clusters Nonhierarchical method that finds based on distance.

Overview of process

Type

- Requires known "K" clusters; Can be difficult to choose.
- Initial cluster choices and order of

Disadvantages

- data strongly affect results.

 Can be difficult to reproduce results due to random initial "centroid"
- Fast (for small k). Easy to implement.

Advantages

- Clusters can be recalibrated.

Big data; Hyper spherical (e.g. 3D sphere).

Works well for

Clustering

Repeatedly links pairs of clusters until every data object is included.

- Initial seeds, data order have strong
- effect.

 Merges cannot be undone.

 No statistical / theoretical foundation for results.

 Sensitive to outliers.

- Easy to implement.

 Dendogram makes for easy visualization of "k".
- Results easily reproducible.

Small to medium data. Performance and execution time increase dramatically for large data sets.

PCA

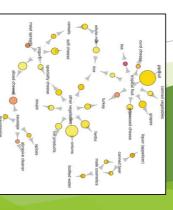
Dimension Reduction

Transforms high dimensional data into low dimensional data using orthogonal transformations.

- Principal components (a linear combination of the original features) can be challenging to interpret and read, compared to the original features.
- Data must be standardized beforehand.
- Good visualization tool.
- Reduces irrelevant or redundant
- features) Reduces overfitting (by reducing features.

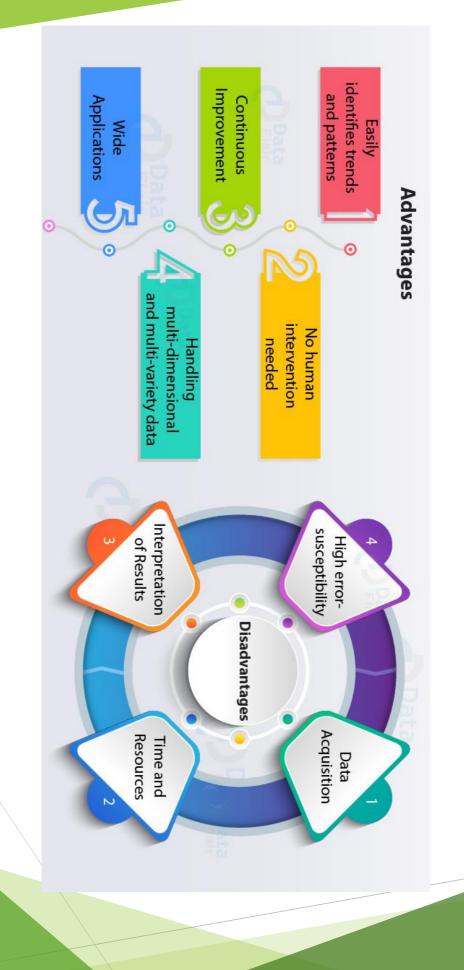
Extracting important features (components) from big data.

Association



21

Machine Learning - Advantages & Disadvantages



Machine Learning - Email spam case study

ABSTRACT

The increasing volume of unsolicited bulk e-mail (also known as spam) has generated a need for reliable anti-spam filters. Machine learning techniques now days used to automatically filter the spam e-mail in a very successful rate. In this paper we review some of the most popular machine learning methods (Bayesian classification, k-NN, ANNS, SVMs, Artificial immune system and Rough sets) and of their applicability to the problem of spam Email classification. The comparison of algorithms performance on the spam. corpus is presented.

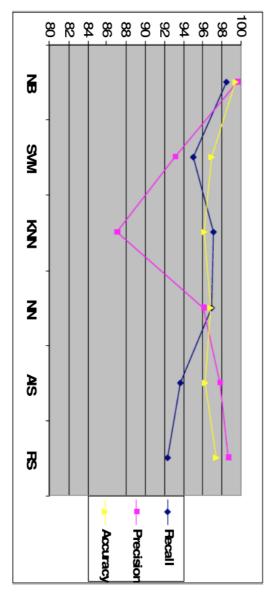


Figure 1. Spam Recall. Spam Precision and Accuracy curves of six classifiers

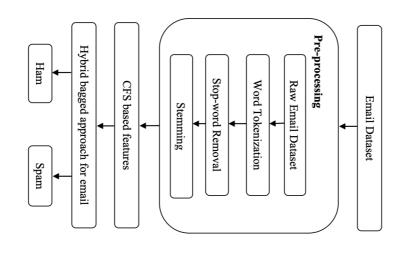
Note :- Naïve bayes method has a very satisfying performance among the other methods

Ref: https://www.researchgate.net/publication/50211017_Machine_Learning_Methods_for_Spam_E-Mail_Classification

Machine Learning - Email spam case study

algorithms. Third experiment is the proposed SMD system implemented using hybrid bagged approach. The overall accuracy of 87.5% achieved by the hybrid bagged approach based SMD system. Abstract:

Spam email is one of the biggest issues in the world of internet. Spam emails not only influence the organisations financially but also exasperate the individual email user. This paper aims to propose a machine learning based hybrid bagging approach by implementing the two machine learning algorithms: Naïve Bayes and J48 (decision tree) for the spam email terms of precision, recall, accuracy, f-measure, true negative rate, false positive rate and false negative rate. The two experiments are performed using individual Naïve Bayes & J48 detection. In this process, dataset is divided into different sets and given as input to each algorithm. Total three experiments are performed and the results obtained are compared in



		** * * * * * * * * * * * * * * * * * * *	:
Evaluation Measures	Naïve Bayes	J48	Hybrid Bagged approach
TP	81	89	58
FP	14	6	10
TN	86	94	06
FN	19	11	15
Precision (%)	85.26	93.68	89.47
Recall (%)	81	89	85
Accuracy (%)	83.5	91.5	87.5
F-Measure (%)	89.27	84.8	87.03
TNR (%)	86	94	90
FPR (%)	19	11	15
FNR (%)	14	6	01

Note :- J48 method has a very satisfying performance among the other methods

Machine Learning - Summary

Semi supervised	Unsupervised Learning	Supervised Learning	
Some data is labeled, some not. Goal: better results than labeled data alone. Good for real world data.	Unlabeled data (inputs only) is analyzed. Learning happens without supervision.	Majority of algorithms. Machine is trained using well-labeled data; inputs and outputs are matched.	Overview
Combination of above processes.	Inputs are used to create a model of the data.	Mapping function takes inputs and matches to outputs, creating a target function.	Process
All the above.	Clustering, Association.	Classification, Regression	Subtypes
Self training, Mixture models, Semi-supervised SVM	PCA, k-Means, Hierarchical clustering.	Linear regression, Random forest, SVM.	Examples

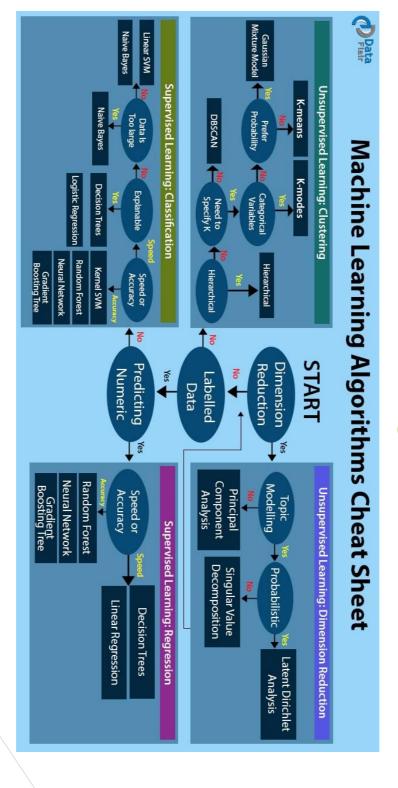
Thank You



Appendix



Prediction model using AIML Forecasting techniques, Building a



Forecasting techniques, Prediction models, choosing

right algorithm with examples Repeated Incremental Pruning to Produce Error Reduction (RIPPER) Least Absolute Shrinkage and Selection Operator (LASSO) Locally Estimated Scatterplot Smoothing (LOESS) Multivariate Adaptive Regression Splines (MARS) Radial Basis Function Network (RBFN) Ordinary Least Squares Regression (OLSR) Gradient Boosted Regression Trees (GBRT) Convolutional Neural Network (CNN) Deep Boltzmann Machine (DBM) Bootstrapped Aggregation (Bagging) Least Angle Regression (LARS) Gradient Boosting Machines (GBM) Stacked Generalization (Blending) Deep Belief Networks (DBN) Stacked Auto-Encoders Back-Propagation Hopfield Network Ridge Regression Stepwise Regression Logistic Regression Zero Rule (ZeroR) Linear Regression One Rule (OneR) Perceptron Elastic Net Random Forest AdaBoost Boosting Neural Networks Deep Learning Regularization Rule System Regression Ensemble Machine Learning Algorithms Instance Based Dimensionality Reduction Decision Tree Bayesian Gaussian Naive Bayes Bayesian Network (BN) Multinomial Naive Bayes Bayesian Belief Network (BBN) Averaged One-Dependence Estimators (AODE) k-Medians Hierarchical Clustering **Expectation Maximization** C5.0 C4.5 3 Chi-squared Automatic Interaction Detection (CHAID) Iterative Dichotomiser 3 (ID3) Locally Weighted Learning (LWL) Conditional Decision Trees Decision Stump Classification and Regression Tree (CART) Self-Organizing Map (SOM) Learning Vector Quantization (LVQ) k-Nearest Neighbour (kNN) Regularized Discriminant Analysis (RDA) Partial Least Squares Discriminant Analysis Multidimensional Scaling (MDS) Linear Discriminant Analysis (LDA) Flexible Discriminant Analysis (FDA) Quadratic Discriminant Analysis (QDA) Mixture Discriminant Analysis (MDA) Principal Component Regression (PCR) Projection Pursuit Sammon Mapping Partial Least Squares Regression (PLSR Principal Component Analysis (PCA)

29