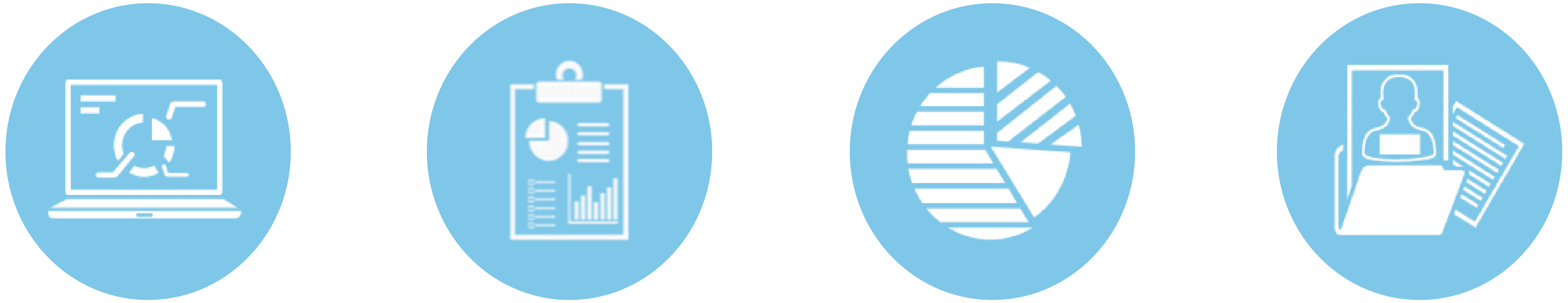
Machine Learning

Lesson 3: Supervised Learning



Concepts Covered

- Types of Supervised Learning
- Various Regression Algorithms
- Advanced Regularization Techniques
- Logistic Regression
- Accuracy Metrics

Learning Objectives

By the end of this lesson, you will be able to:

- Understand the different types of supervised learning
- Build various regression models

Supervised Learning

Topic 1: Overview

Supervised Learning

Supervised Learning is a type of machine learning used to train models from labeled training data. It allows you to predict output for future or unseen data.

Examples of Supervised Learning

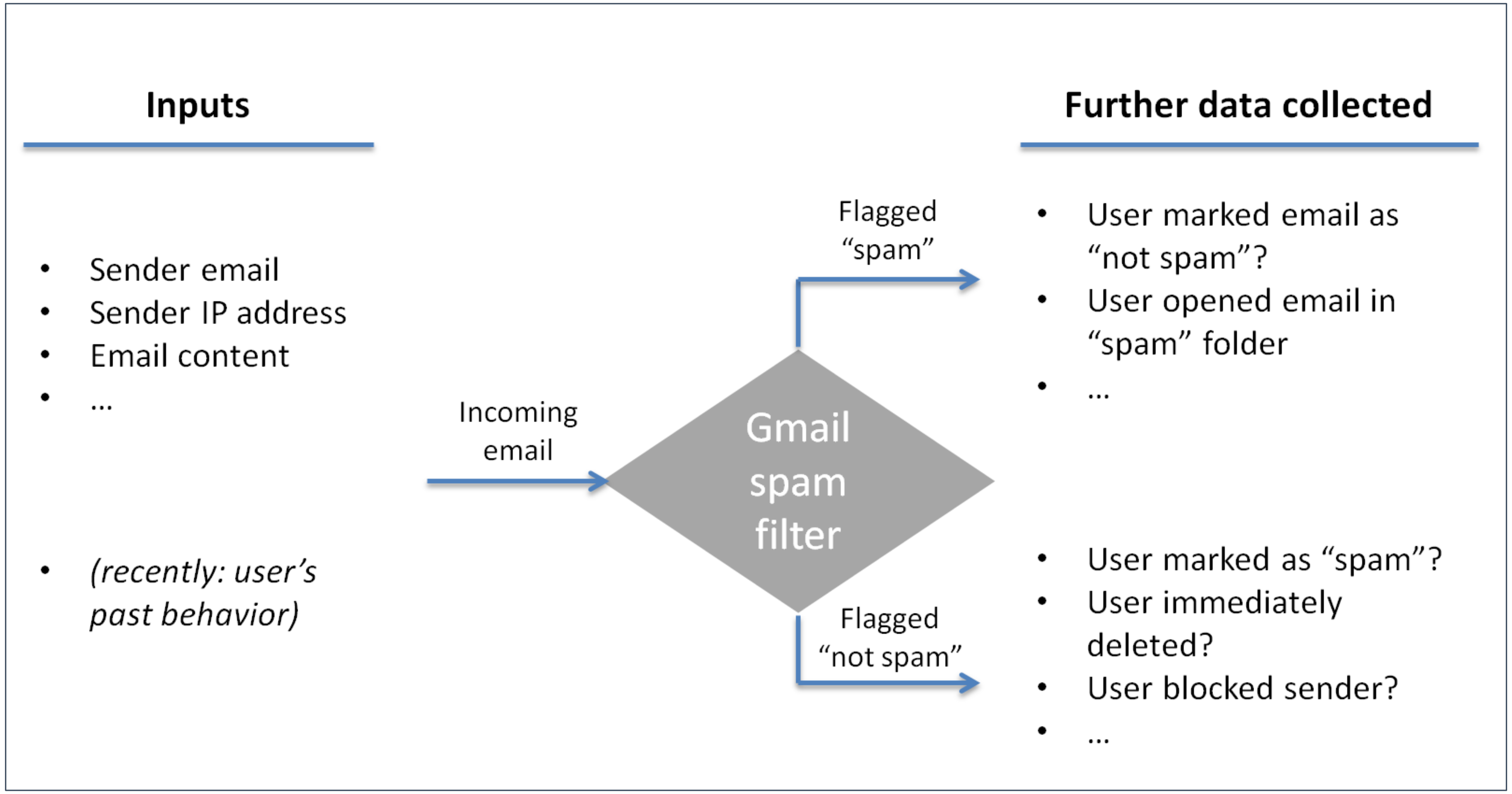
Example 1: Weather Apps

The predictions made by weather apps at a given time are based on prior knowledge and analysis of weather over a period of time for a particular place.

Examples of Supervised Learning (Contd.)

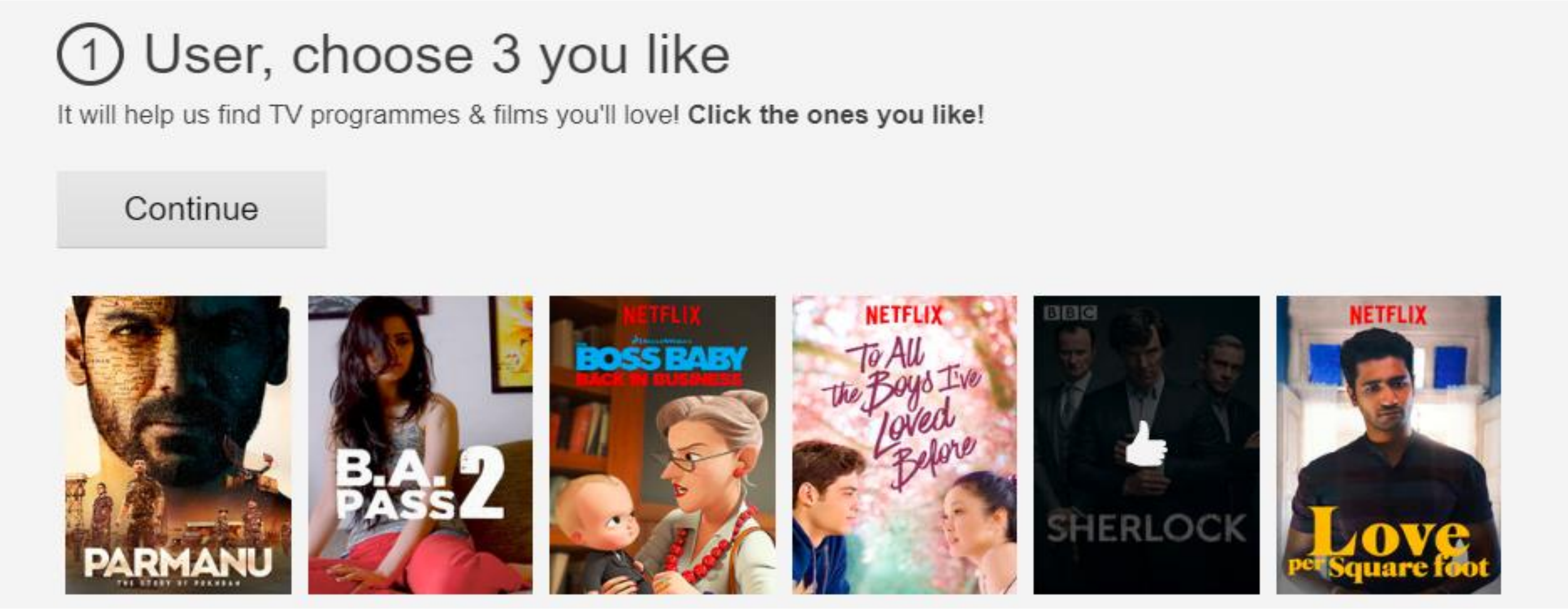
Example 2: Gmail Filters

Gmail filters, a new email into Inbox (normal) or Junk folder (Spam) based on past information of spam.



Real-Life Scenario

Ever wondered how Netflix makes recommendations?



Supervised Learning: Case Study

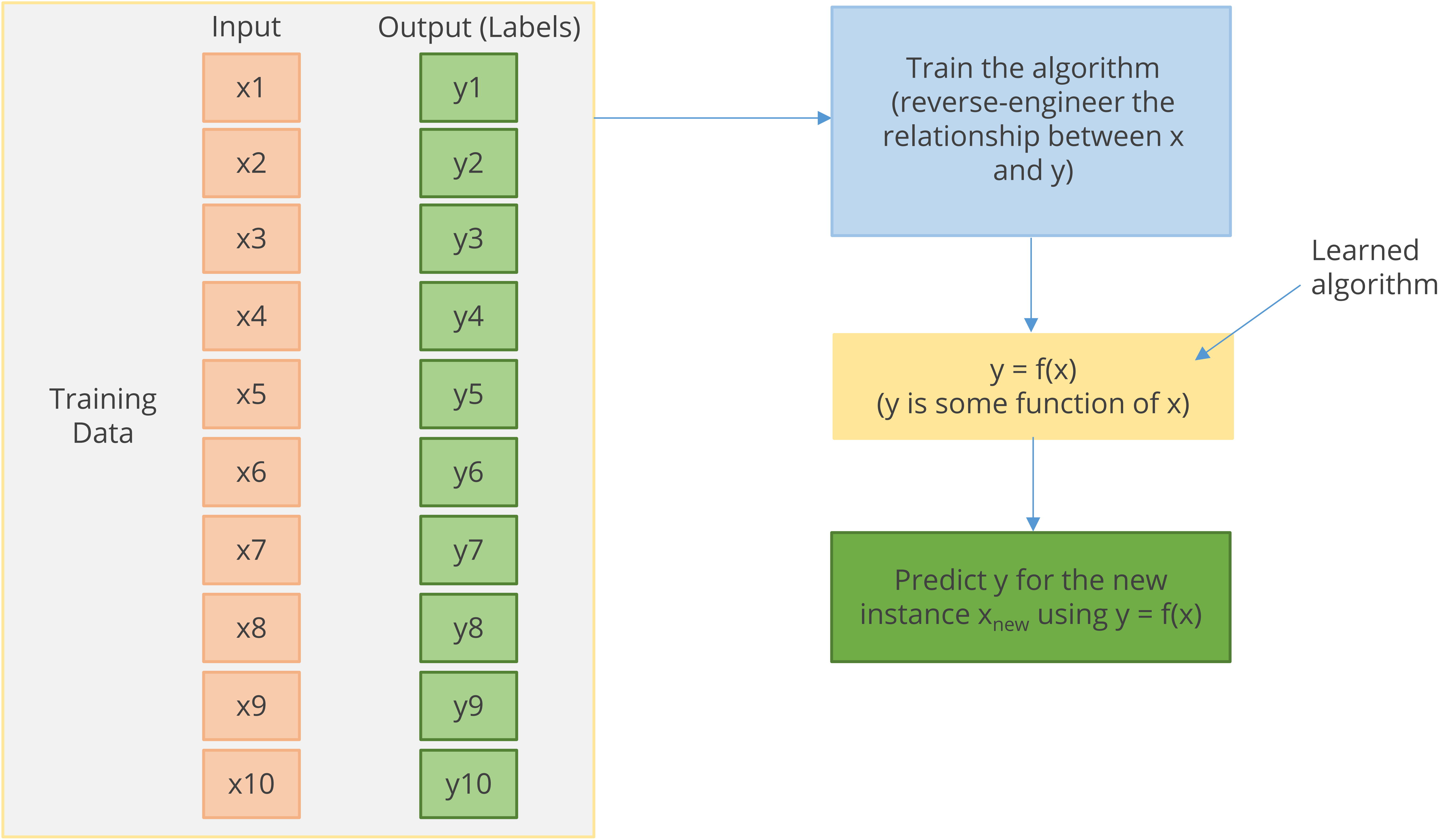
Netflix uses **supervised learning** algorithms to recommend users the shows they may watch based on the viewing history and ratings by similar classes of users

New Input

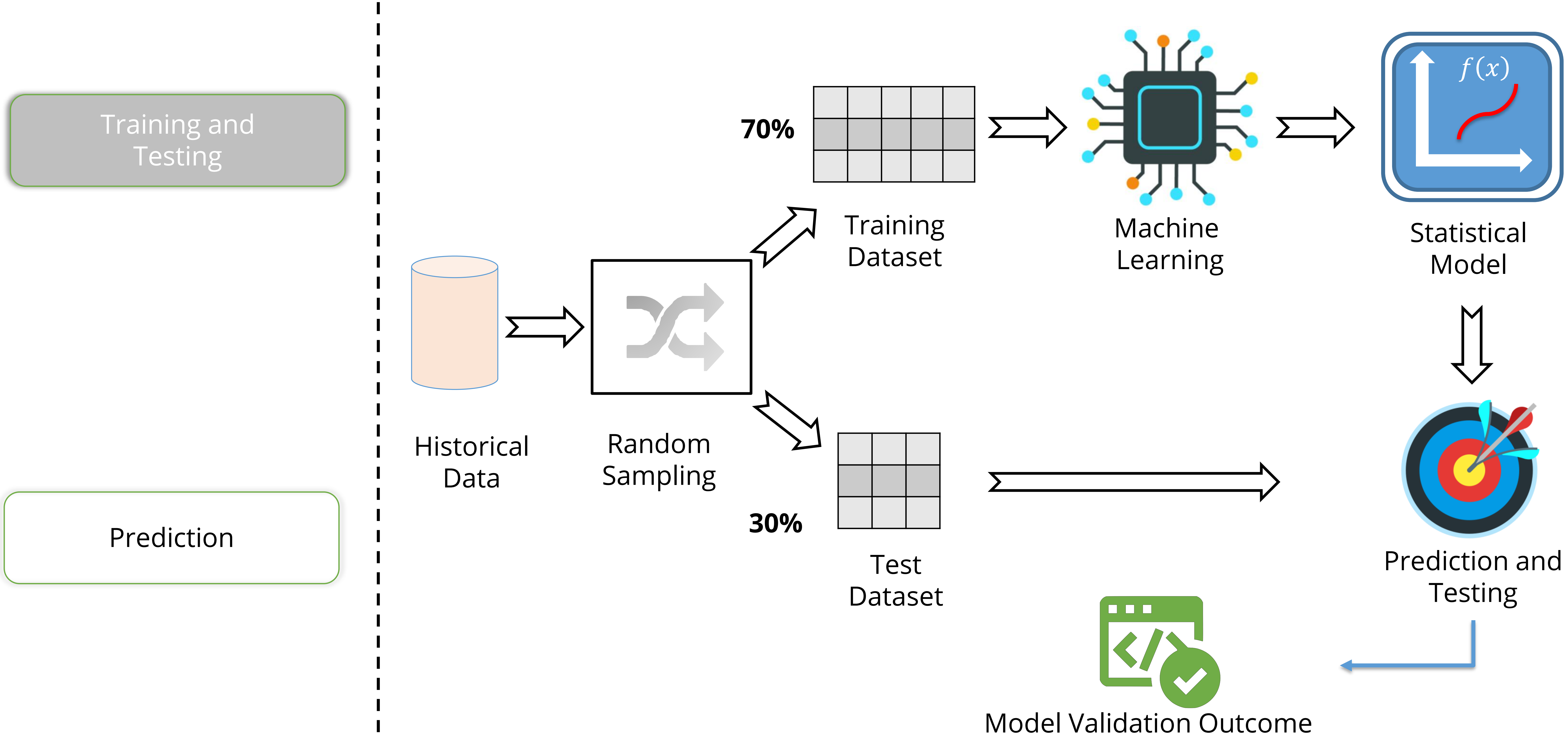
Algorithm Trained on Historical Data

Predicted Outcome

Understanding the Algorithm



Supervised Learning Flow

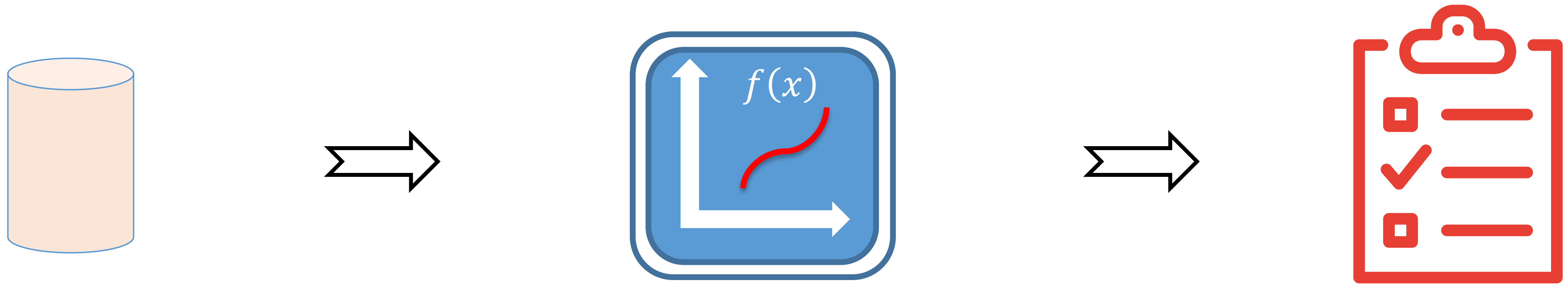


Supervised Learning Flow

Training and Testing

Prediction

Use the learned algorithm y = f(x) to predict production data



New Data

Model

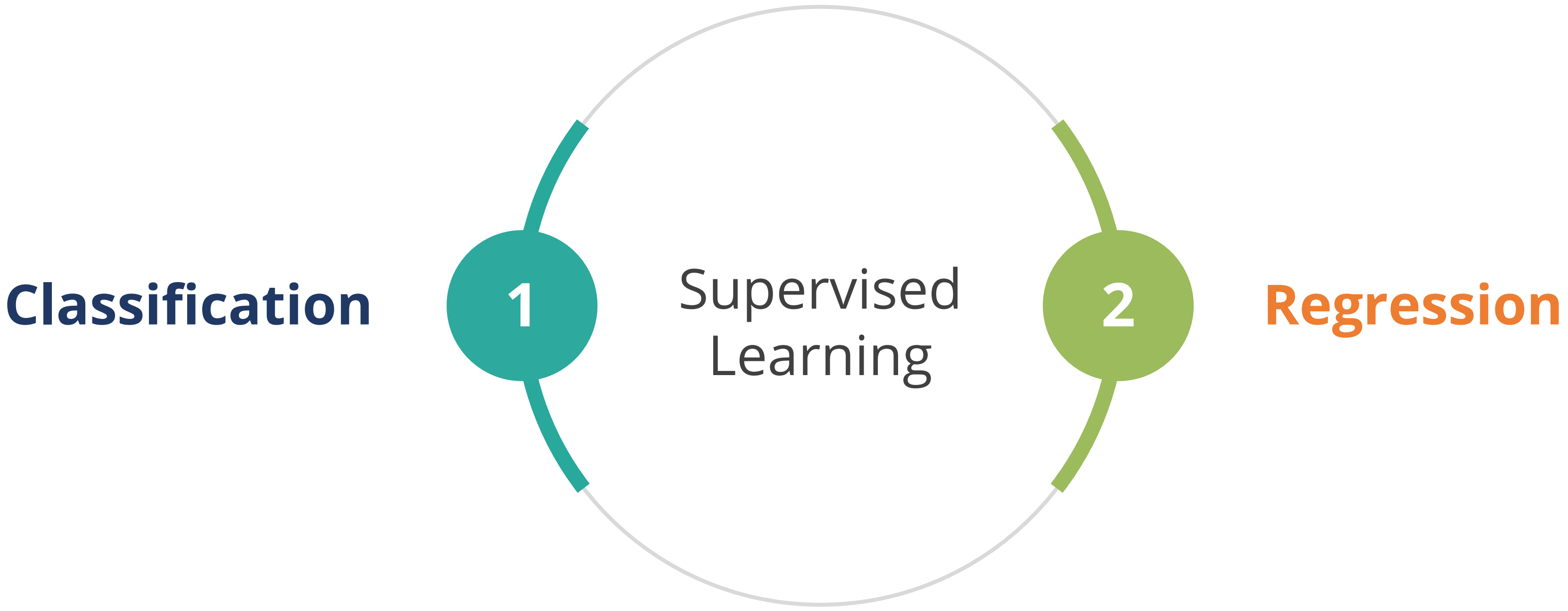
Prediction Outcome

Algorithm prediction can be improved by more training data, capacity, or algorithm redesign.

Supervised Learning

Topic 2: Types of Supervised Learning

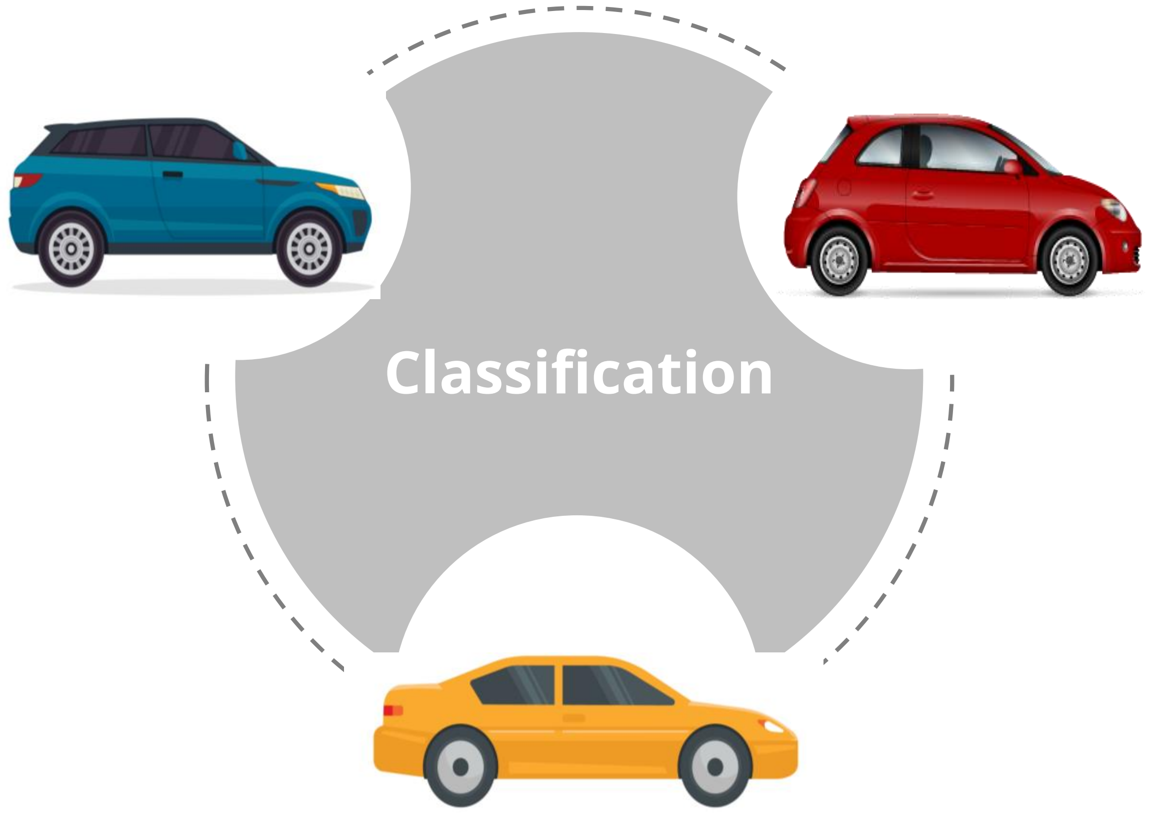
Types of Supervised Learning



In supervised learning, algorithm is selected based on target variable.

Types of Supervised Learning (Contd.)

If target variable is categorical (classes), then use classification algorithm.



In other words, classification is applied when the output has finite and discreet values.

Example: Predict the class of car given its features like horsepower, mileage, weight, colour, etc.

The classifier will build its attributes based on these features.
Analysis has three potential outcomes - Sedan, SUV, or Hatchback

Types of Supervised Learning (Contd.)

If target variable is a continuous numeric variable (100–2000), then use a regression algorithm.

Example: Predict the price of a house given its sq. area, location, no of bedrooms, etc.

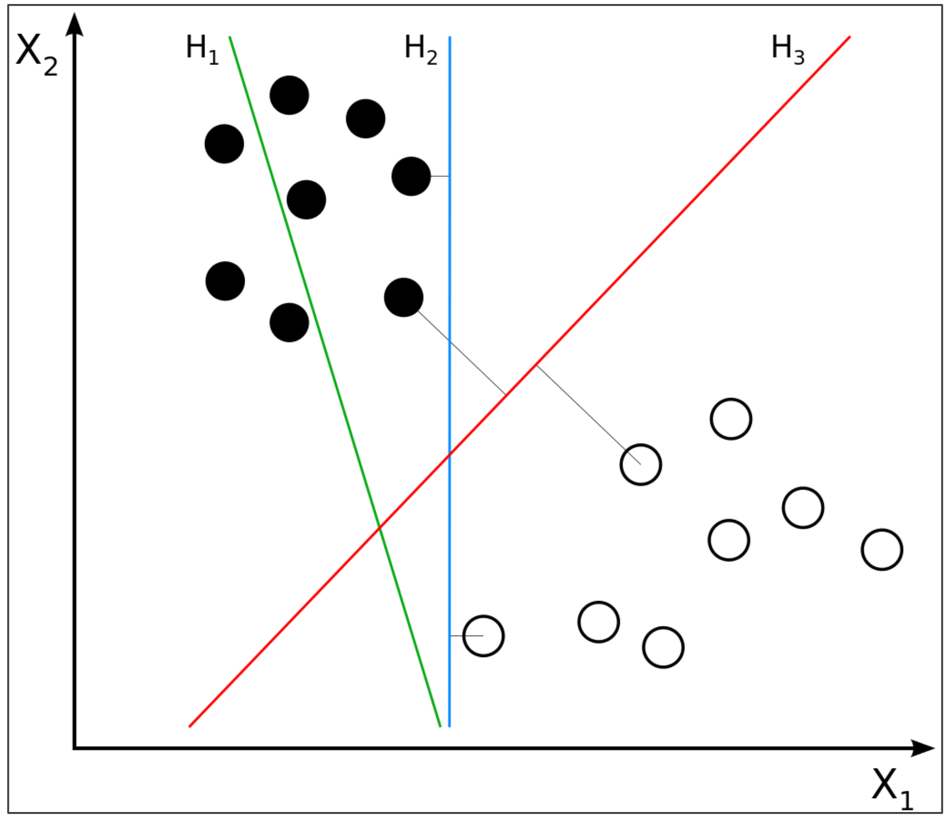
A simple regression algorithm is given below

$$y = w * x + b$$

This shows relationship between price (y) and sq. area (x) where price is a number from a defined range.

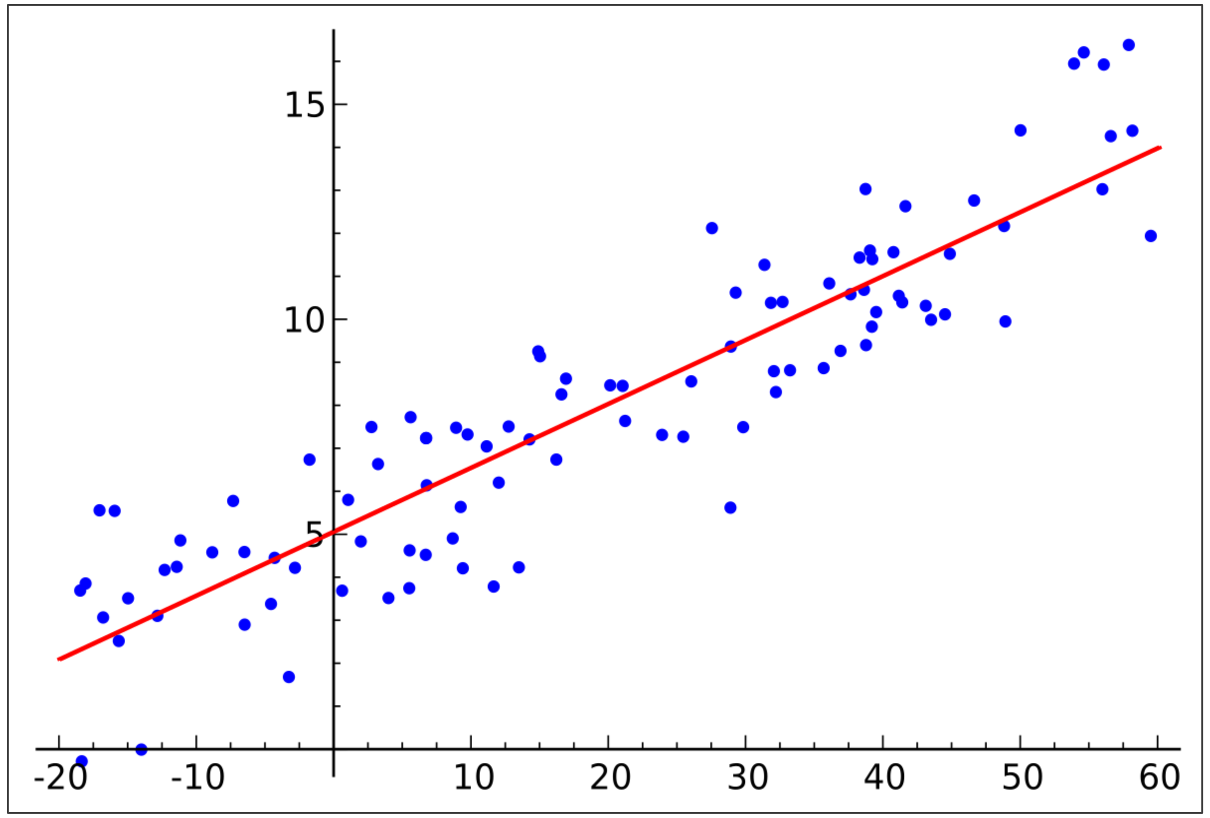
Types of Supervised Learning (Contd.)

Classification



What Class ?

Regression



How much ?

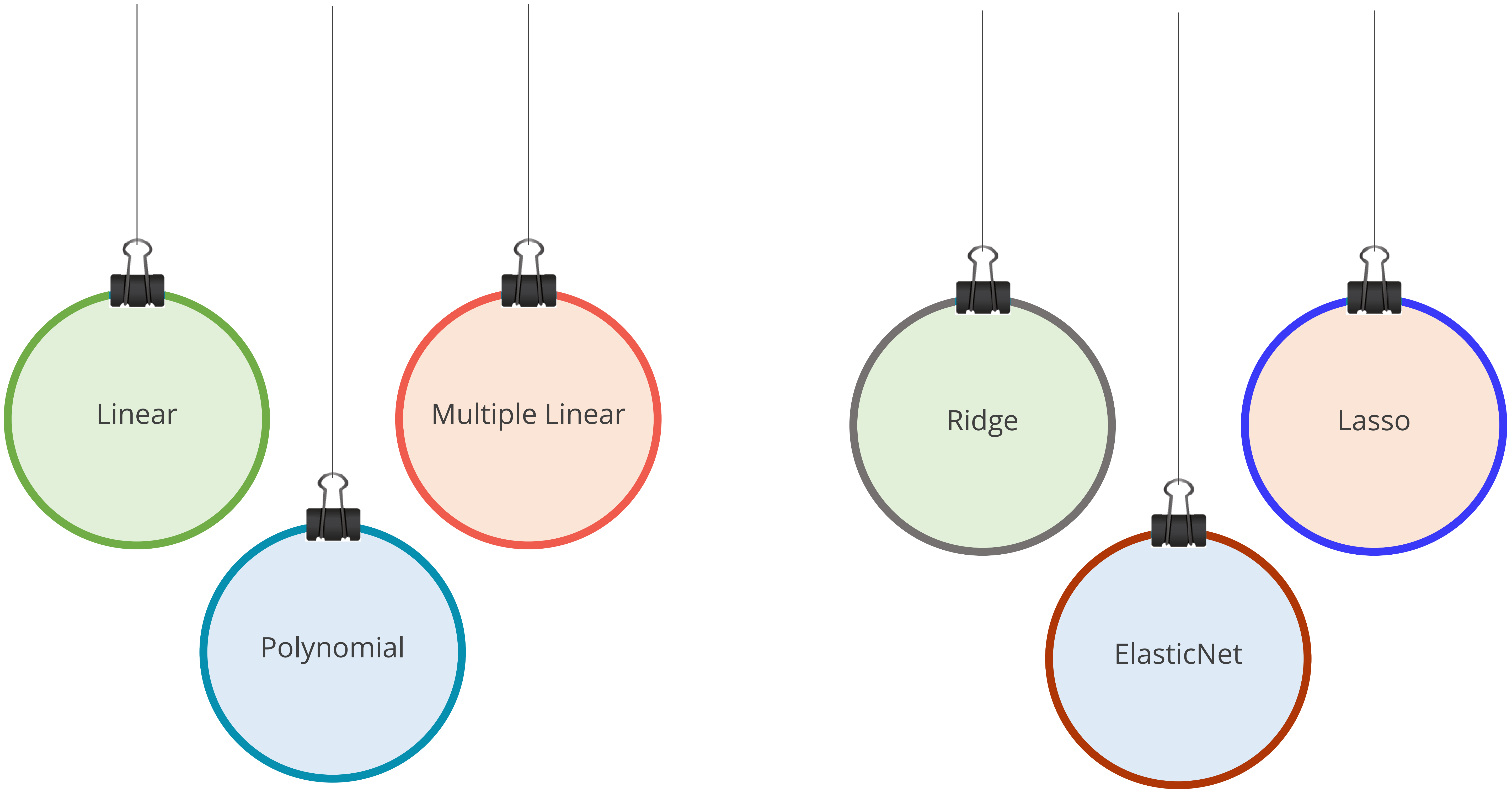
Types of Classification Algorithms

Logistic Regression	Used to estimate discrete values (binary values like 0/1, yes/no, true/false) based on given set of independent variable(s)
Decision Trees	Decision Trees make sequential, hierarchical decisions about the outcome variable based on the predictor data
Random Forest	Random Forest is an ensemble of decision trees. It gives better prediction and accuracy than decision tree
Naive Bayes Classifier	Based on Bayes theorem and works with an assumption that features are independent
Support Vector Machines	SVM draws hyperplane in a feature space that separates instances into different categories with margins in between as far apart as possible

Supervised Learning

Topic 3: Types of Regression Algorithms

Types of regression algorithms



Types of Regression Algorithms

- **Linear Regression**
- Multiple Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

Linear Regression is a statistical model used to predict the relationship between independent and dependent variables denoted by x and y respectively

Examine 2 factors

1

How closely are x and y related ?

Linear regression gives a number between -1 and 1 indicating the strength of correlation between the two variables

0 : no correlation
1 : positively correlated
-1 : negatively correlated

2

Prediction

When the relationship between x and y is known, use this to predict future values of y for a value of x

This is done by fitting a regression line and represented by a linear equation:

$$y = a * x + b$$

Types of Regression Algorithms (Contd.)

- Linear Regression
- **Multiple Linear Regression**
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

Multiple linear regression is a statistical technique used to predict the outcome of a response variable through several explanatory variables and model the relationships between them.

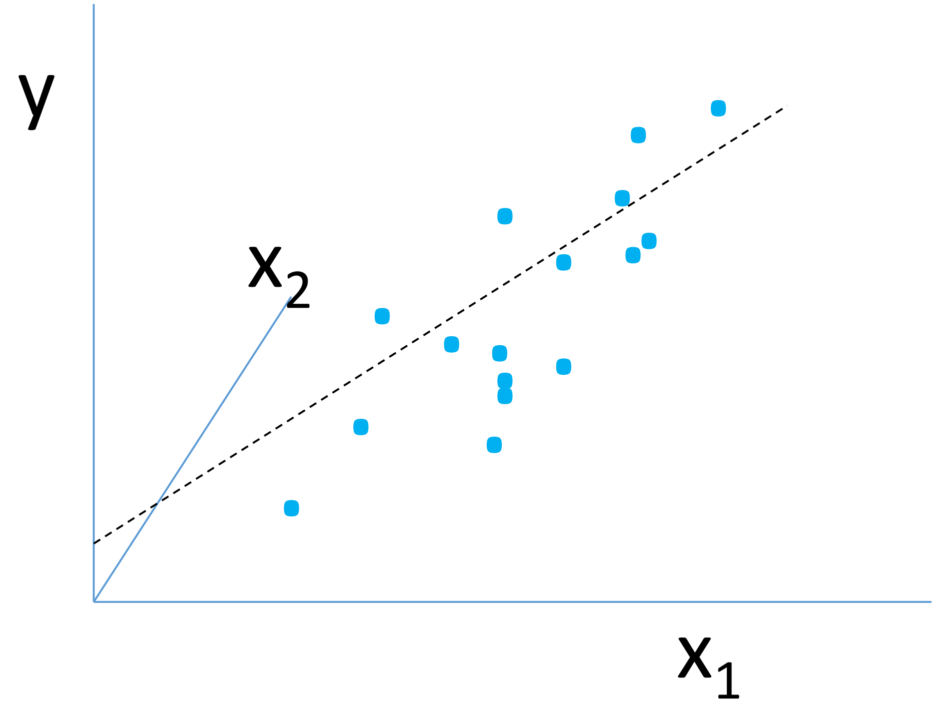
Equation for MLR

$$Y = m_1 * x_1 + m_2 * x_2 + m_3 * x_3 + \ldots\ldots\ldots + m_n * x_n + c$$

Y → Dependent Variable

$m1, m2, m3 \ldots. m_n$ → Slopes

c → Coefficient



Types of Regression Algorithms (Contd.)

- Linear Regression
- Multiple Linear Regression
- **Polynomial Regression**
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

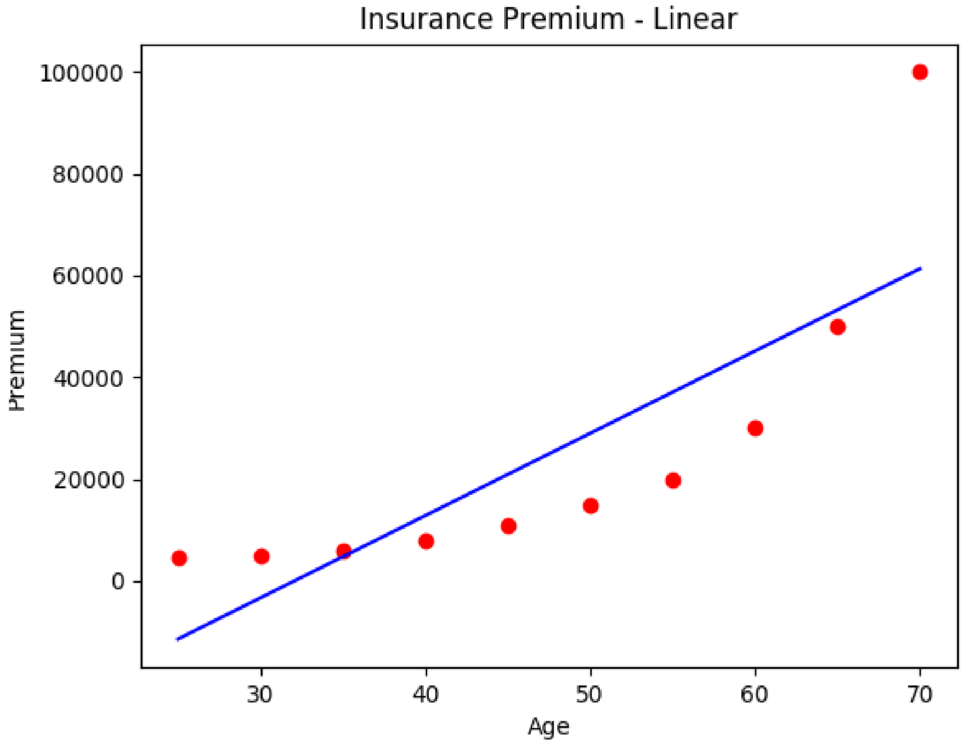
Polynomial regression is applied when data is not formed in a straight line.
It is used to fit a linear model to non-linear data by creating new features from powers of non-linear features.

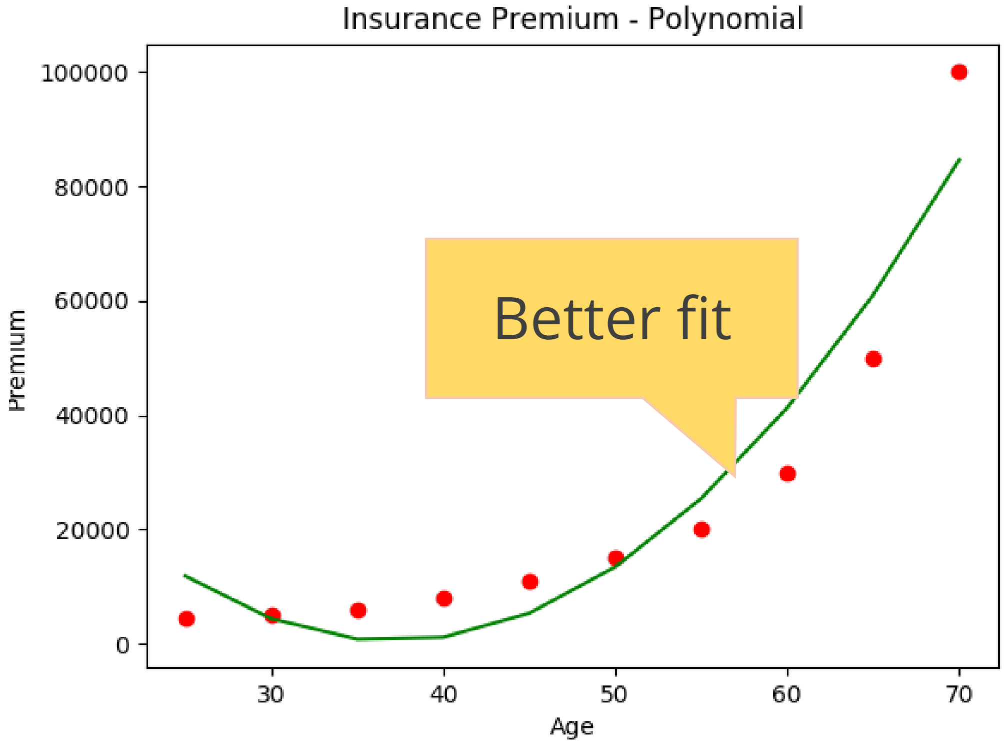
Example: Quadratic features

$$x_2' = x_2^2$$

$$y = w_1x_1 + w_2x_2^2 + 6$$

$$= w_1x_1 + w_2x_2' + 6$$

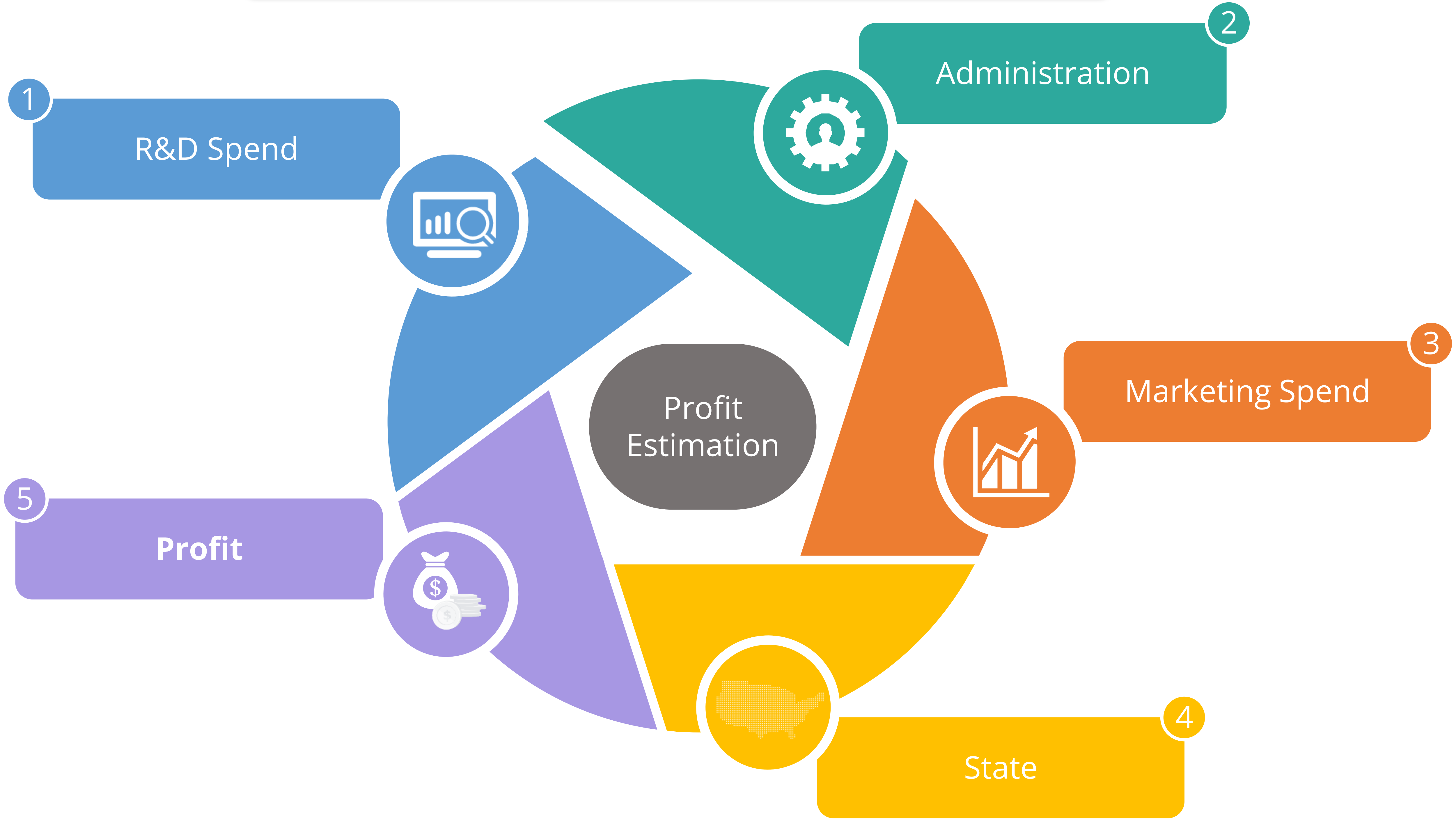




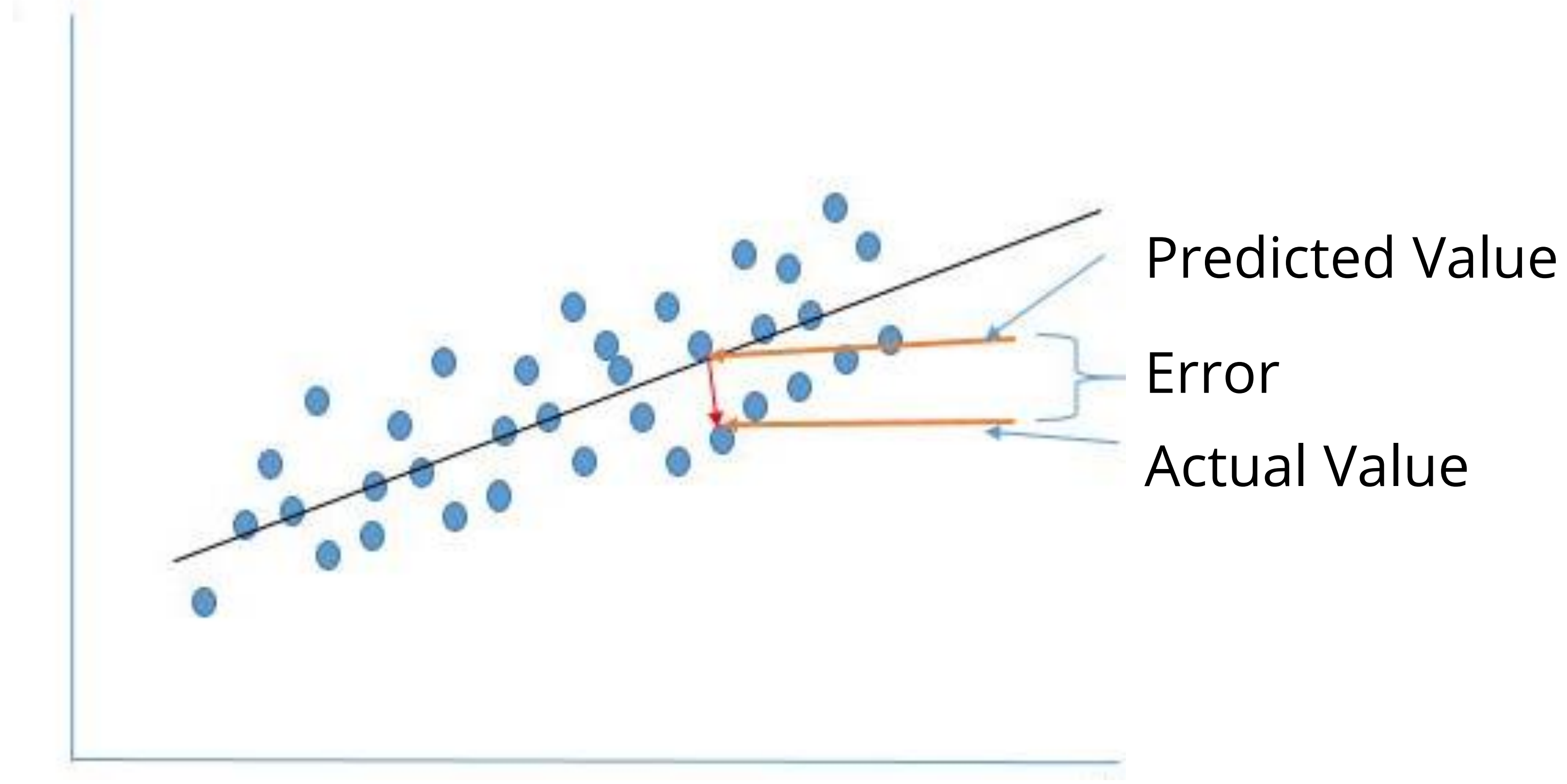
simplilearn

Regression Use Case

Predicting profit based on expenditures of the company



Accuracy Metrics



$$\textbf{R-square} = 1 - \frac{\sum(Y\_actual - Y\_predicted)^2}{\sum(Y\_actual - Y\_mean)^2}$$

R-square is the most common metric to judge the performance of regression models

R^2 lies between 0 -100 %

Example: Performing linear regression on sq. Area (x) and Price (y) returns **R-square** value as 16
This means you have 16% information to make an accurate prediction about the price.

Adjusted R-Squared

The disadvantage with R-squared is that it assumes every independent variable in the model explains variations in the dependent variable.

Use adjusted R-squared when working on a multiple linear regression problem.

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N-1)}{N-p-1}$$

where R^2 is R-squared value

P is number of predictor variables

N is number data points

Cost Function

Mean-Squared Error (MSE) is also used to measure the performance of a model.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$$

Where *N* is the number of data points

y_i is the predicted value by the model

$\bar{y}_i$ is the actual value for the data point

These functions are called the loss function or the cost function, and the value has to be minimized.

Gradient Descent

Gradient descent is another algorithm used to reduce the loss function.

It is an optimization algorithm that tweaks it's parameters (coefficients) iteratively to minimize a given cost function to its minimum.

Model stops learning when the gradient (slope) is zero

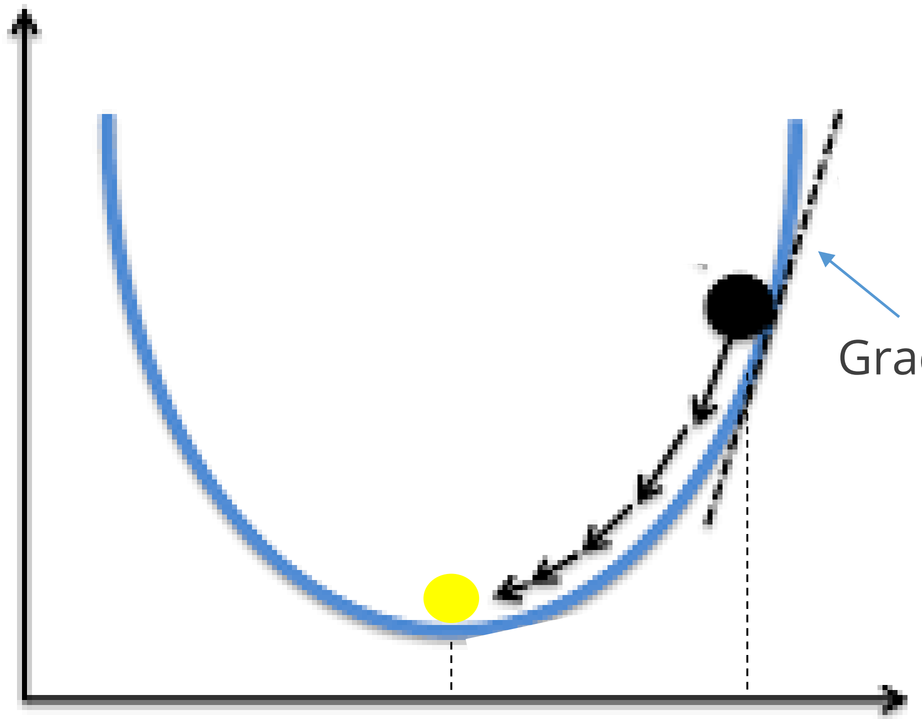
Algorithm :

1) Initialize parameter by some value

2) For each iteration calculate the derivative of the cost function and simultaneously update the parameters until a global minimum

$$\theta := \theta - \alpha \frac{\delta}{\delta\theta} J(\theta)$$

where ἀ is the learning rate



simplilearn

Evaluating Coefficients

In regression analysis, p-values and coefficients together indicate which relationships in the model are statistically significant and the nature of those relationships.

Coefficients describe the mathematical relationship between each independent variable and the dependent variable.

p-values for the coefficients indicate whether these relationships are statistically significant.

$p < 0.05$	**REJECT** the Null hypothesis, meaning variables have some effect and need to be retained
$p > 0.05$	**ACCEPT** the Null hypothesis, meaning variables have no effect and can be removed

Assisted Practice

Regression

Duration: 20 mins.

Problem Statement: The Advertising dataset captures sales revenue generated with respect to advertisement spends across multiple channels like radio, tv, and newspaper.

Objective:

Build a linear regression model to:

- Interpret the coefficients of the model
- Make predictions
- Find and analyze model residuals
- Evaluate model efficiency using RMSE and R-Square values

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Unassisted Practice

Regression

Duration: 20 mins.

Problem Statement: A real estate company wants to build homes at different locations in Boston.

They have data for historical prices but haven't decided the actual prices yet. They want to price it so that it is affordable to the general public.

Objective:

- Import the Boston data from sklearn and read the description using DESCR
- Analyze the data and predict the approximate prices for the houses

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Answer

Code

Importing libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Importing Dataset

```
from sklearn.datasets import load_boston
boston = load_boston()
boston.keys()
boston.feature_names
boston.target
print(boston.DESCR)

df = pd.DataFrame(boston.data)
df.shape
```

Answer (Contd.)

Code

Creating train and test dataset

```
x_train = df.drop(['HOUSING_VALUE'], axis=1)
y_train = df['HOUSING_VALUE’]

xtrain,xtest,ytrain,ytest =
model_selection.train_test_split(x_train,y_train,test_size=0.3,random_state=42)
```

Model Building

```
from sklearn.linear_model import LinearRegression
model = LinearRegression(n_jobs = -1)
model.fit(xtrain,ytrain)

print(model.intercept_)
print(model.coef_)
print(df.columns.values.tolist())
list(zip(df.columns,model.coef_))
```

Answer (Contd.)

Code

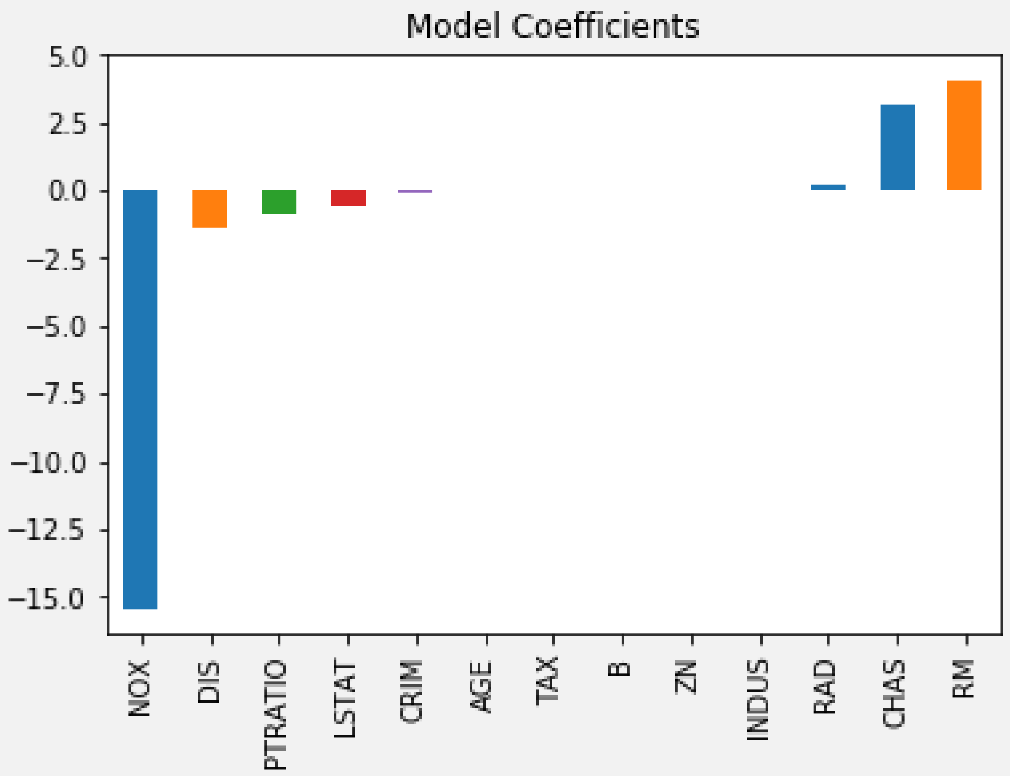
Checking the magnitude of coefficients

```
predictors = df.columns[:-1]
coef = pd.Series(model.coef_,predictors).sort_values()
coef.plot(kind='bar', title='Modal Coefficients')

plt.scatter(df.RM,df.HOUSING_VALUE)
plt.title("Relationship between RM and Target Variable")
plt.show()

plt.scatter(df.NOX,df.HOUSING_VALUE)
plt.title("Relationship between NOX and Target Variable")
plt.show()

print('R2 Value/Coefficient of Determination: {}'.format(model.score(xtest, ytest)))
```



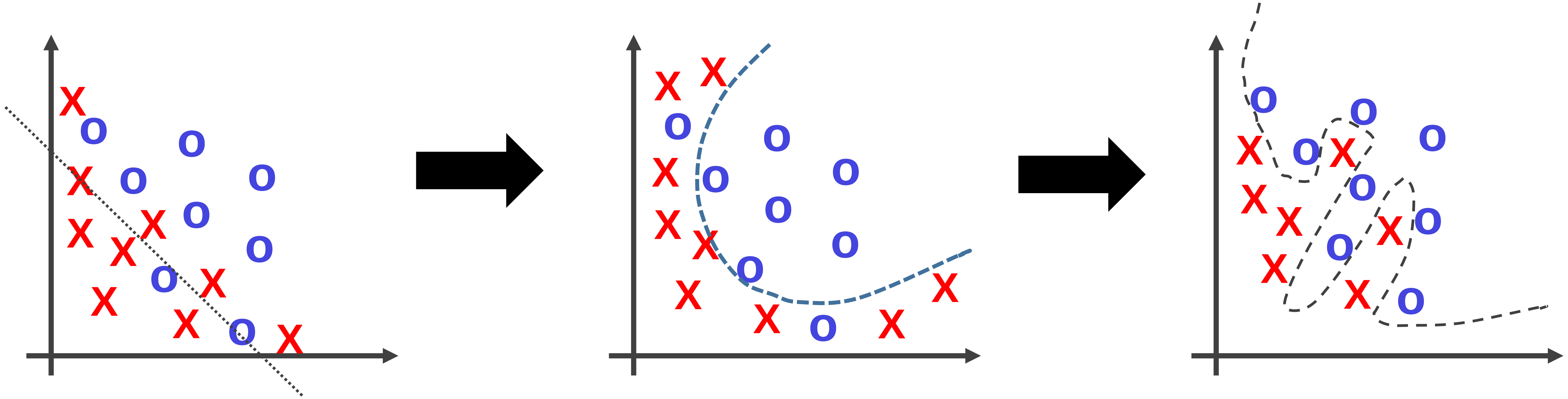
Answer (Contd.)

Code

Final prediction

```
plt.scatter(ytrain,model.predict(xtrain))
print(sqrt(mean_squared_error(ytrain, model.predict(xtrain))))
plt.scatter(ytest,model.predict(xtest))
print(sqrt(mean_squared_error(ytest, model.predict(xtest))))
pd.DataFrame({'Actual': ytest, 'Predicted': model.predict(xtest)}).head(10)
```

Challenges in Prediction



Under Fit | Appropriate | Over Fit

If the model learning is poor, you have an **underfitted** situation

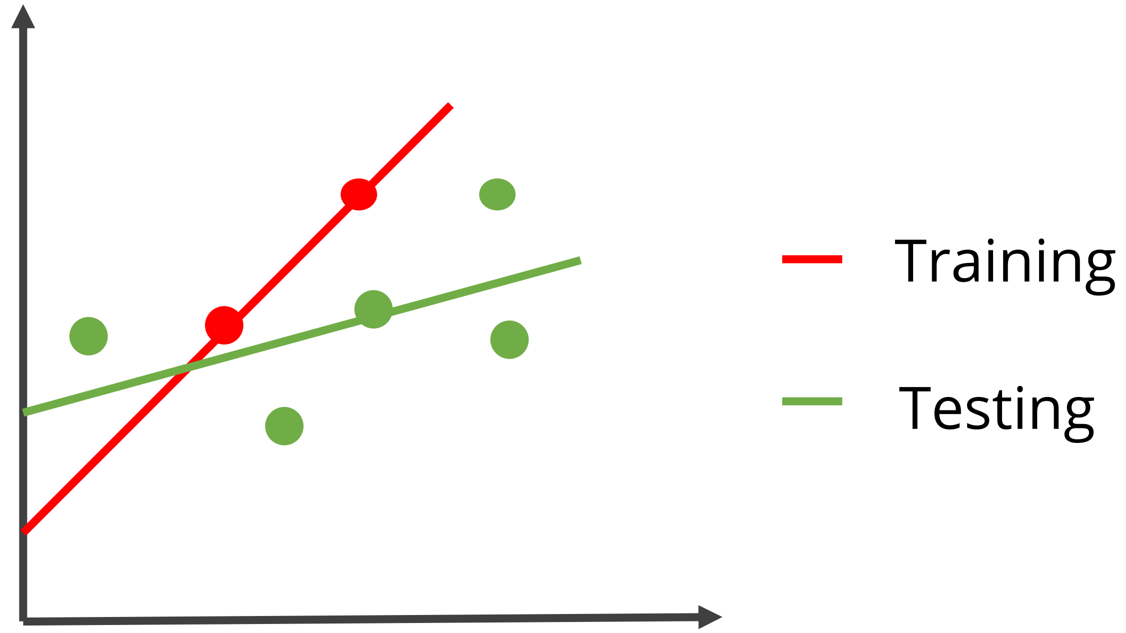
The algorithm will not work well on test data
Retraining may be needed to find a better fit

Overfitting happens when model accuracy for training data is good, but model does not generalize well to the overall population

Algorithm is not able to give good predictions for the new data

Regularization

Regularization solves overfitting to the training data.



Used to restrict the parameters values that are estimated in the model

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2$$

This loss function includes 2 elements.

1) the sum of square distances between predicted and actual value

2) the second element is the regularization term

Types of Regression (Contd.)

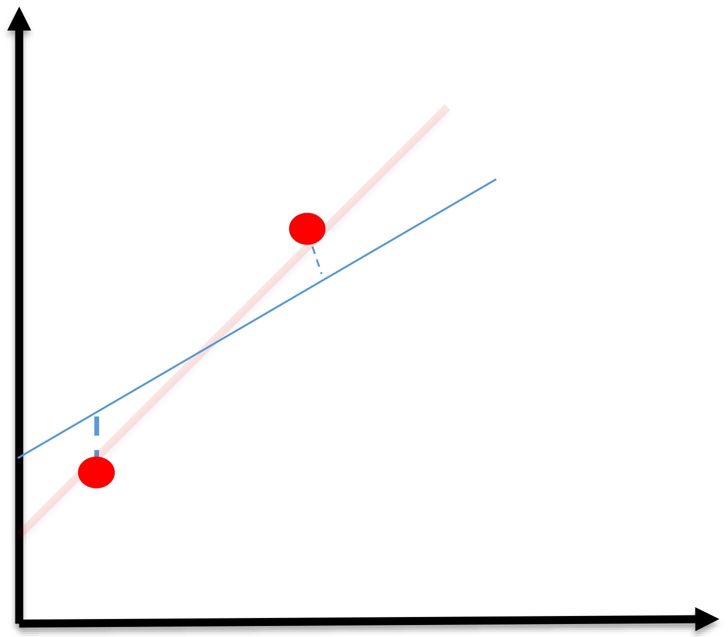
- Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- **Ridge Regression**
- Lasso Regression
- ElasticNet Regression

Ridge Regression (L2) is used when there is a problem of multicollinearity.

By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

The main idea is to find a new line that has some bias with respect to the training data

In return for that small amount of bias, a significant drop in variance is achieved



$$\boxed{\text{Minimization objective} = \text{LS Obj} + \lambda * (\text{sum of the square of coefficients})}$$

LS Obj refers to least squares objective

λ controls the strength of the penalty term

Types of Regression (Contd.)

- Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

Lasso Regression (L1) is similar to ridge, but it also performs feature selection.

It will set the coefficient value for features that do not help in decision making very low, potentially zero.

Minimization objective = LS Obj + λ * (sum of absolute coefficient values)

Lasso regression tends to exclude variables that are not required from the equation, whereas ridge tends to do better when all variables are present.

simplilearn

Types of Regression (Contd.)

- Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression

ElasticNet regression combines
the strength of **lasso and ridge regression**

$$\boxed{\begin{gathered}\text{the sum of the squared residuals}\\ +\\ \underbrace{\lambda_1 \times |variable_1| + \dots + |variable_x|}_{\text{Lasso penalty}} \;+\; \underbrace{\lambda_2 \times variable_1^2 + \dots + variable_x^2}_{\text{Ridge penalty}}\end{gathered}}$$

If you are not sure whether to use lasso or ridge, use ElasticNet

Assisted Practice

Regression

Duration: 20 mins.

Problem Statement: BigMart has collected sales data for 1559 products across 10 stores in different cities. Attributes of each product and store have been defined.

Objective:

- Build a predictive model and find out the sales of each product at a particular store
- Using Ridge and Lasso regression techniques, interpret the coefficients of the model
- Make predictions using the model
- Evaluate model efficiency using RMSE and R-Square values

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Unassisted Practice

Regression

Duration: 10 mins.

Problem Statement: For the Boston dataset used earlier, the team also wants to cross-reference results using regularization techniques

Objective:

- Build a predictive model using Ridge, Lasso and ElasticNet
- Compare the models basis on accuracy

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Answer

Code

Ridge

```
from sklearn.linear_model import Ridge
ridgeReg = Ridge(alpha=0.001, normalize=True)
ridgeReg.fit(xtrain,ytrain)
print(sqrt(mean_squared_error(ytrain, ridgeReg.predict(xtrain))))
print(sqrt(mean_squared_error(ytest, ridgeReg.predict(xtest))))
print('R2 Value/Coefficient of Determination: {}'.format(ridgeReg.score(xtest, ytest)))
```

Lasso

```
from sklearn.linear_model import Lasso
lassoreg = Lasso(alpha=0.001, normalize=True)
Lasso.fit(xtrain,ytrain)
print(sqrt(mean_squared_error(ytrain, Lasso.predict(xtrain))))
print(sqrt(mean_squared_error(ytest, Lasso.predict(xtest))))
print('R2 Value/Coefficient of Determination: {}'.format(Lasso.score(xtest, ytest)))
```

Answer (Contd.)

Code

ElasticNet

```
from sklearn.linear_model import ElasticNet
Elastic = ElasticNet(alpha=0.001, normalize=True)
Elastic.fit(xtrain,ytrain)
print(sqrt(mean_squared_error(ytrain, Elastic.predict(xtrain))))
print(sqrt(mean_squared_error(ytest, Elastic.predict(xtest))))
print('R2 Value/Coefficient of Determination: {}'.format(Elastic.score(xtest,
ytest)))
```

Supervised Learning

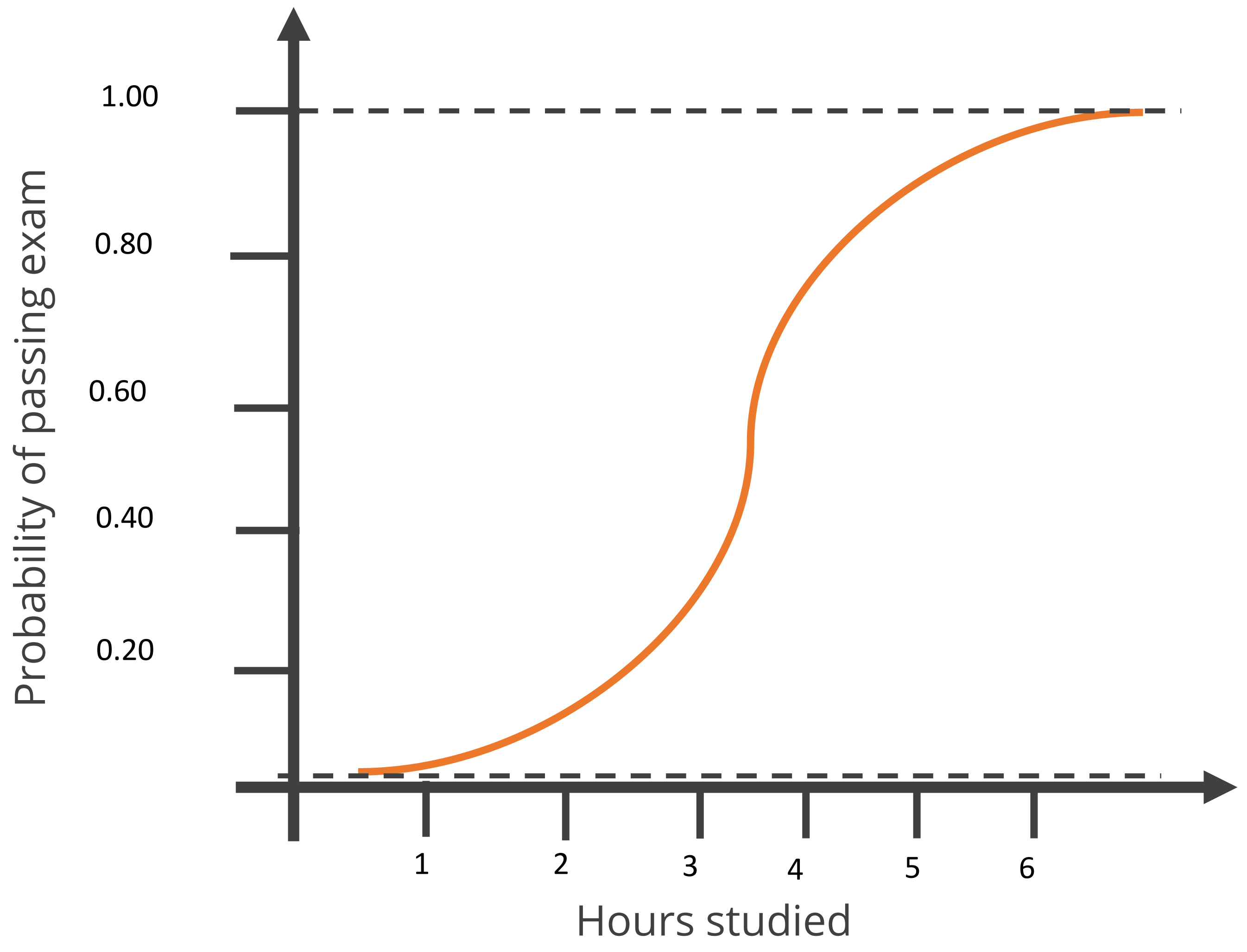
Topic 4: Logistic Regression

Logistic Regression

Logistic Regression is widely used to predict binary outcomes for a given set of independent variables.

The dependent variable's outcome is discrete such as $y \in \{0, 1\}$

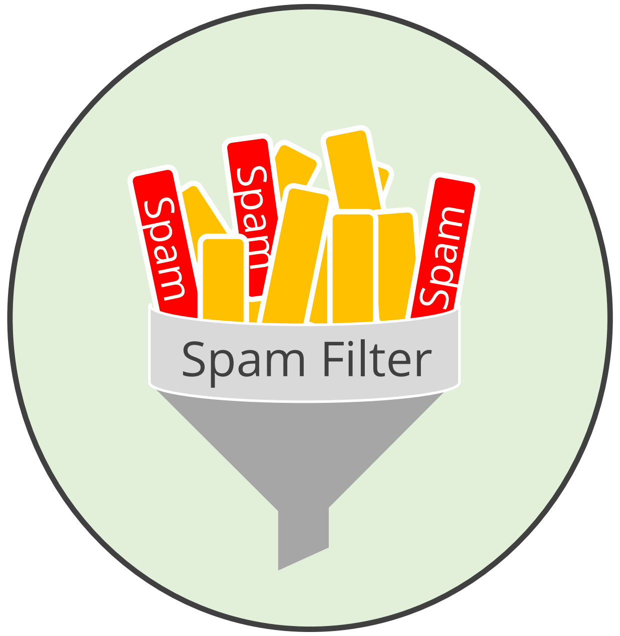
A binary dependent variable can have only two values such as 0 or 1, win or lose, pass or fail, healthy or sick.



Use Cases

Loan sanction

Customer segments



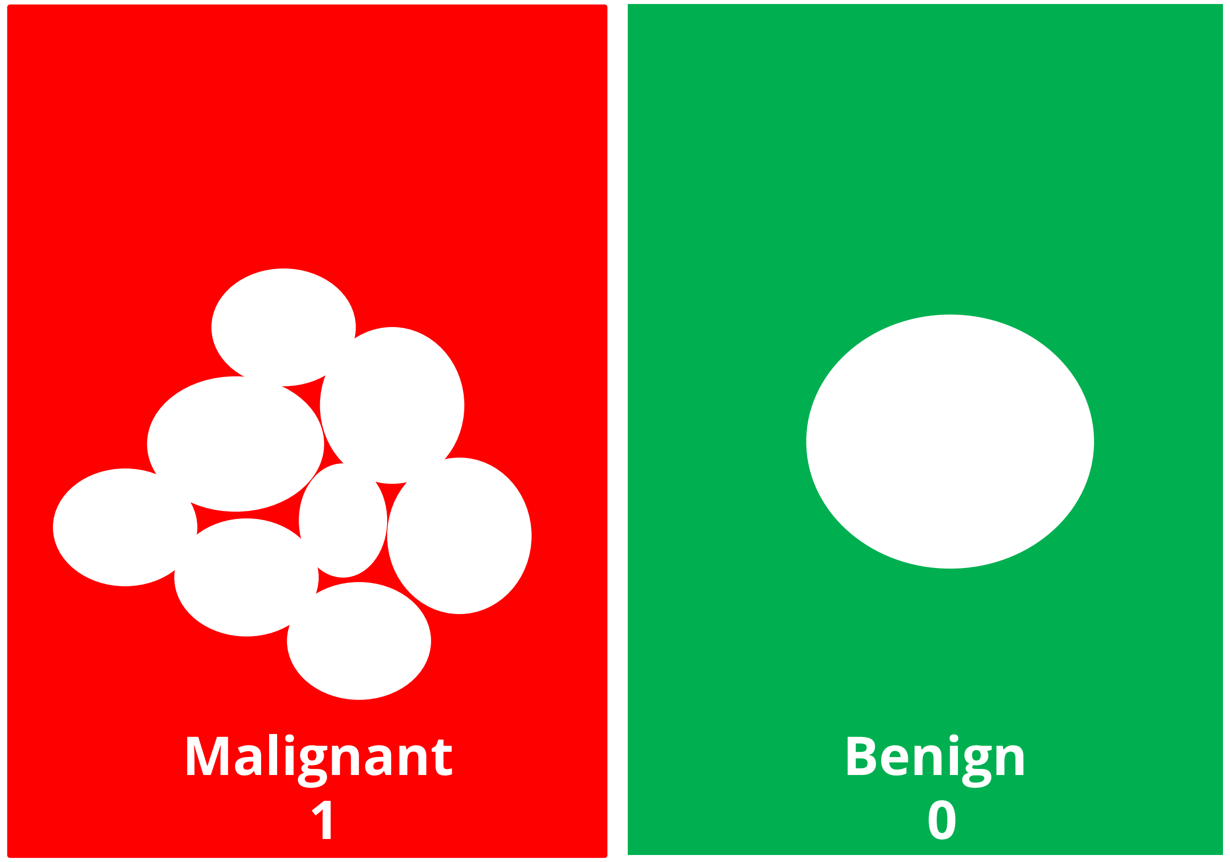
Spam filtering

Exam results – Pass/Fail

Real-Life Scenarios

Weather Forecast

sunny, stormy, cloudy, rainy



Cancer Prediction

Malignant (cancerous) and Benign (non-cancerous)

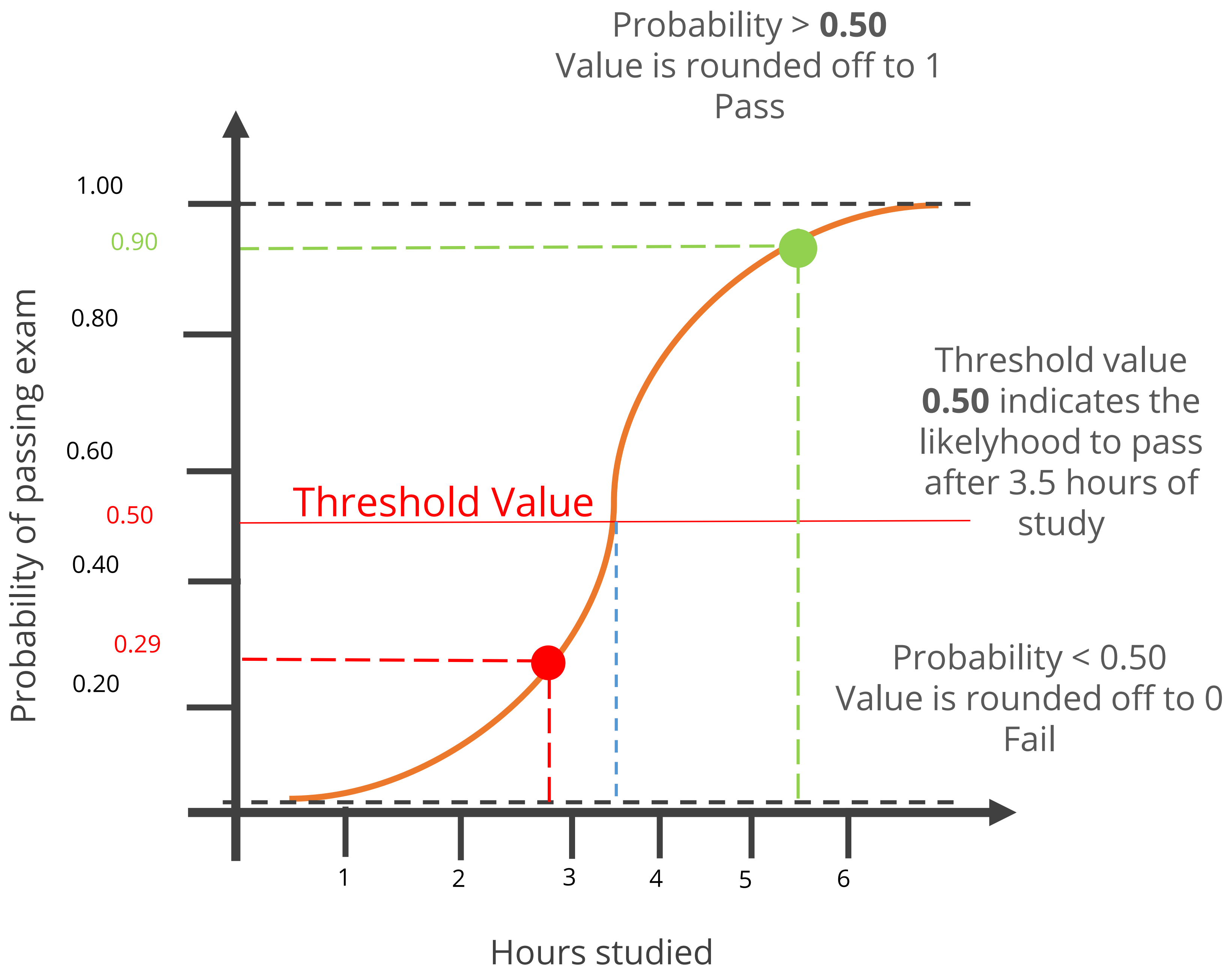
Logistic Regression (Contd.)

The probability distribution of output y is restricted to 1 or 0. This is called as **sigmoid probability (σ)**

If σ (θTx) > 0.5, set y = 1, else set y = 0.

Unlike Linear Regression (and its Normal Equation solution), there is no closed form solution for finding optimal weights of Logistic Regression.

Instead, you must solve this with **maximum likelihood estimation** (a probability model to detect maximum likelihood of something happening).



Logistic Regression Equation

The Logistic regression equation is derived from the straight line equation:

Equation of a straight line

$Y = bx1 + cx2 + D$ $\longrightarrow$ Range is from – (infinity) to (infinity)

Deducing the logistic regression equation from straight line equation

$Y = bx1 + cx2 + D$ $\longrightarrow$ In logistic equation, Y can be only from 0 to 1

Transform it to get the range

$$\frac{Y}{1-Y} \quad \left.\begin{array}{l} Y= 0 \text{ then } 0 \\ Y= 1 \text{ then infinity} \end{array}\right\} \longrightarrow \text{Now, the range is between 0 to infinity}$$

Transform it further to get range: (infinity) to (infinity)

$$\boxed{\log\left[\frac{Y}{1-Y}\right] \Rightarrow Y = bx1 + cx2 + D} \longrightarrow$$

Final Logistic Regression Equation

Sigmoid Probability

The probability in the logistic regression is represented by the Sigmoid function (logistic function or the S-curve).

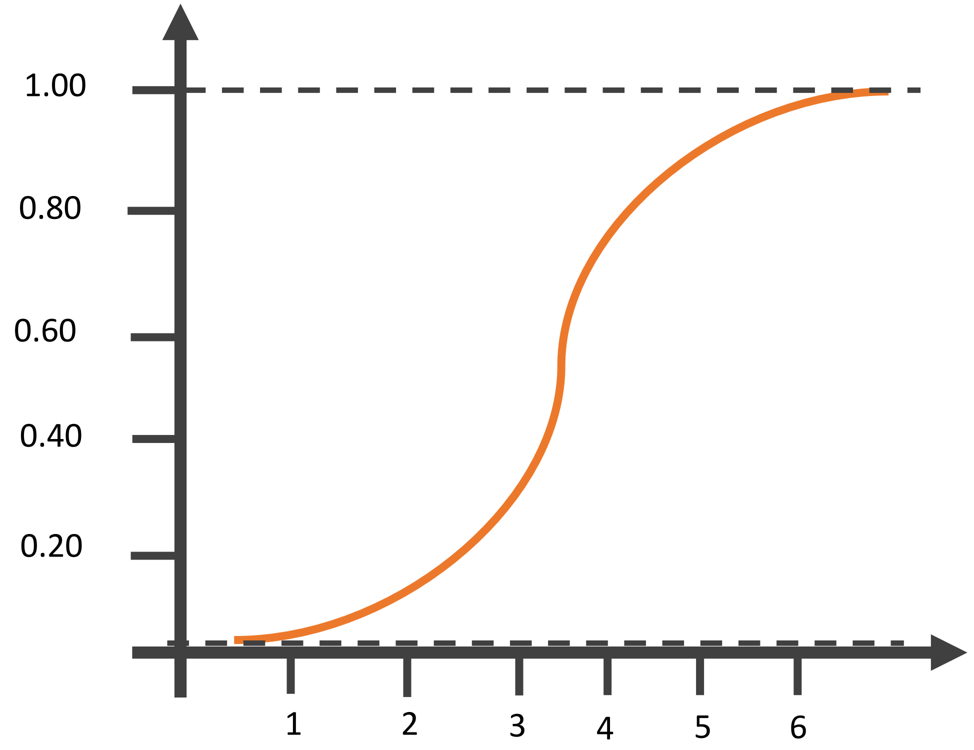
$$S(t) = \frac{1}{1 + e^{-t}}$$

t represents data values * number of hours studied
S(t) represents the probability of passing the exam.

The sigmoid function gives an 'S' shaped curve.

This curve has a finite limit that is Y can only be 0 or 1

0 as x approaches to $-\infty$
1 as x approaches to $+\infty$



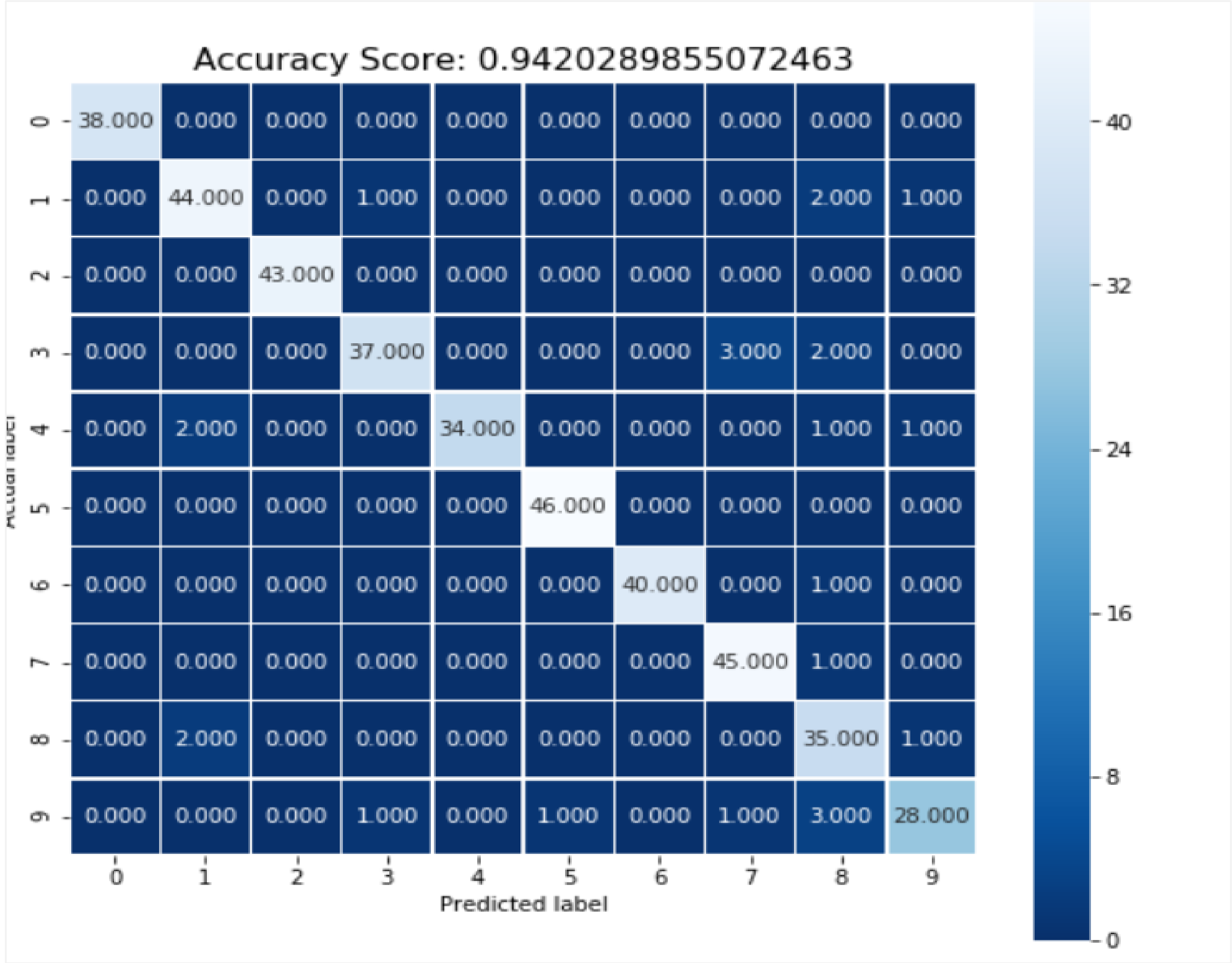
Accuracy Metrics

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix

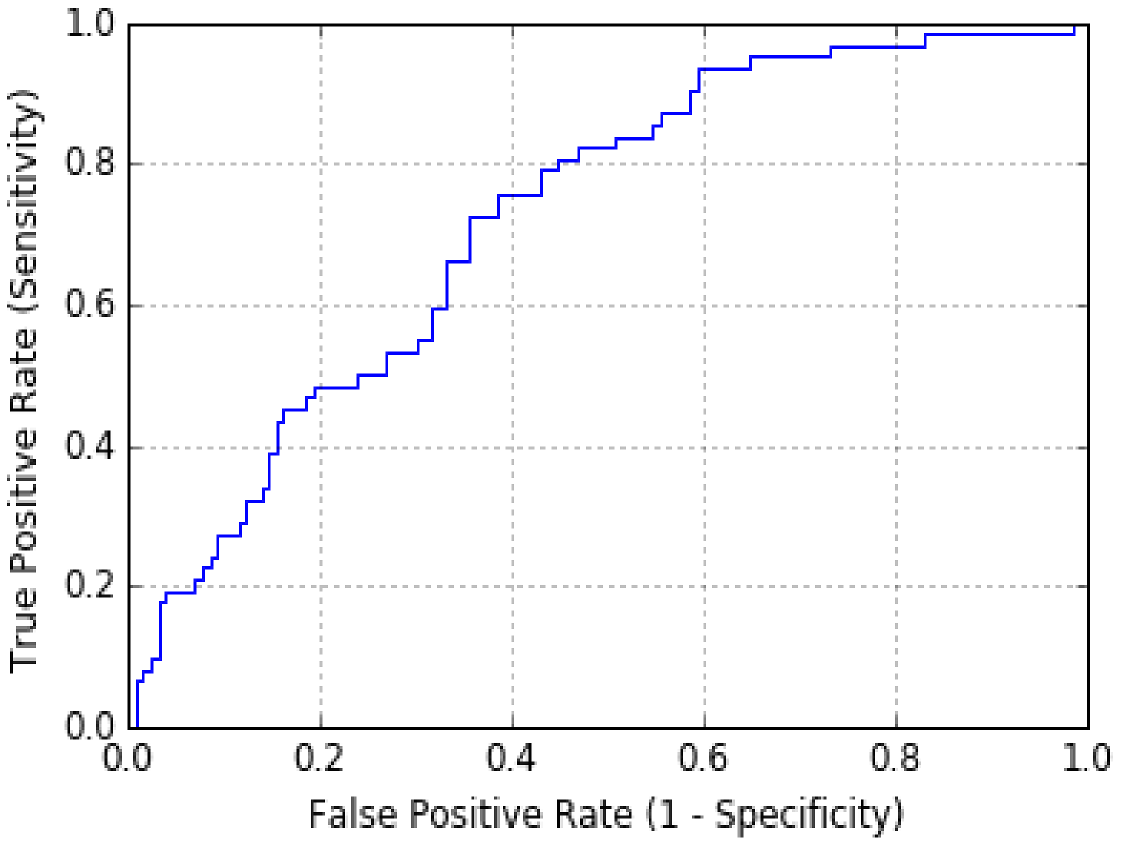
$$\textbf{Accuracy} = \frac{\text{True Positive (TP) + True Negative (TN)}}{\text{Total}}$$

Confusion Matrix for a multi - class classification



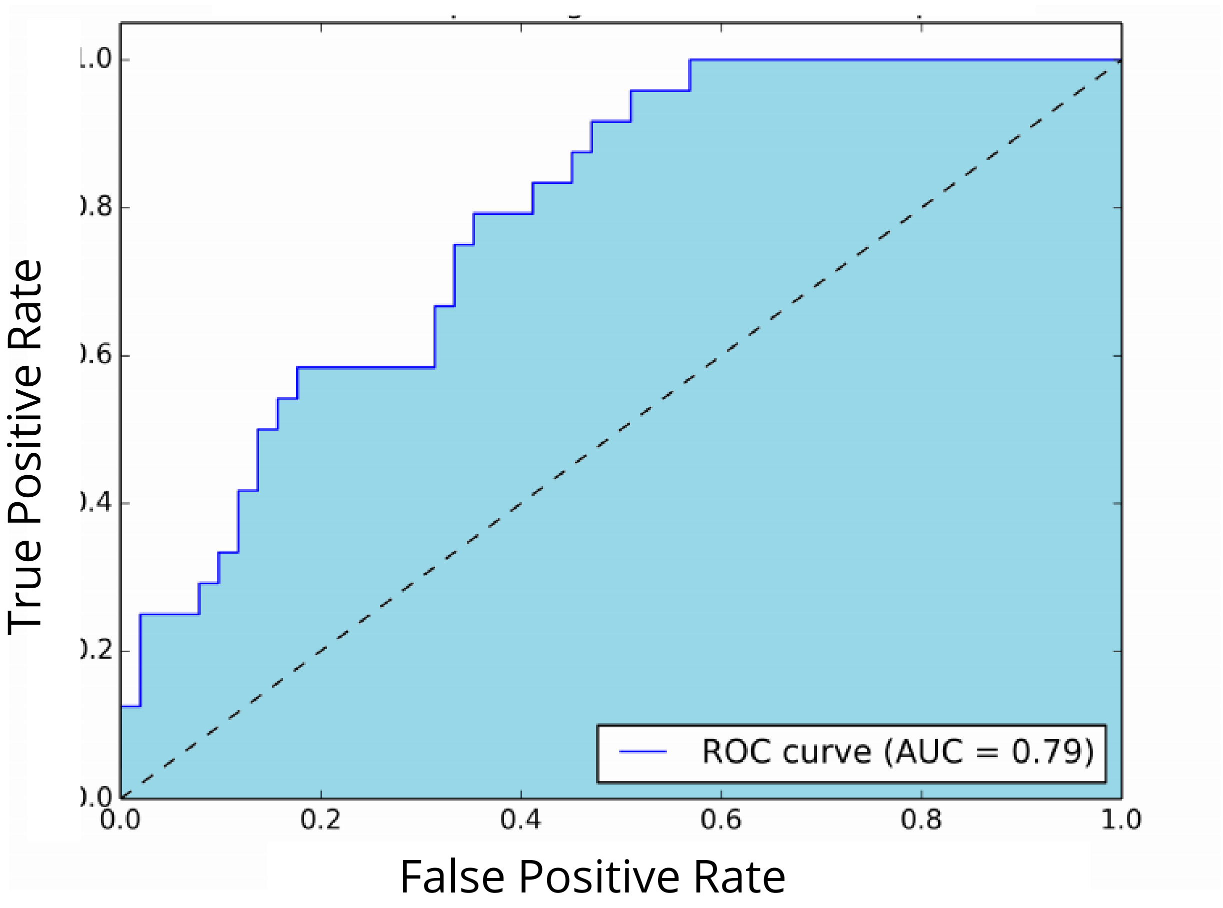
simplilearn

Accuracy Metrics (Contd.)



ROC curve

Compares the model true positive and false positive rates to the ones from a random assignment



AUC (Area under the ROC Curve)

Measures the entire two-dimensional area under the entire ROC curve

Assisted Practice

Regression

Duration: 20 mins.

Problem Statement: The sinking of the Titanic is one of the biggest maritime disaster in the history, killing 1502 out of 2224 passengers and the crew. One of the reasons for such loss was that there were not enough lifeboats. Some groups of people were more likely to survive than others, such as women, children, and the upper-class.

Objective:

- Use logistic regression to predict the survival of a given passenger based on features, such as sex, age

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Unassisted Practice

Regression

Duration: 20 mins.

Problem Statement: The Iris plant has 3 species - Iris Setosa, Iris Versicolour, Iris Virginica
One class is linearly separable from the other two; the latter are not linearly separable from each other.

Objective:

- Import the iris dataset using sklearn
- Use logistic regression to predict the class of iris plant

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Answer

Code

Importing the required libraries

```
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
import matplotlib.colors
from sklearn.linear_model import LogisticRegression
```

Importing the dataset

```
iris = datasets.load_iris()
X = iris.data[:, [2, 3]]
y = iris.target

print('Class labels:', np.unique(y))
```

Answer (Contd.)

Code

Train test split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)

print('Labels counts in y:', np.bincount(y))
print('Labels counts in y_train:', np.bincount(y_train))
print('Labels counts in y_test:', np.bincount(y_test))
```

Standardizing features

```
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
```

Answer (Contd.)

Code

Plotting decision surface

```
def plot_decision_regions(X, y, classifier, test_idx=None, resolution=0.02):
    # setup marker generator and color map
    markers = ('s', 'x', 'o', '^', 'v')
    colors = ('red', 'blue', 'lightgreen', 'gray', 'cyan')
    cmap = matplotlib.colors.ListedColormap(colors[:len(np.unique(y))])

    # plot the decision surface
    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution),np.arange(x2_min, x2_max,
resolution))
    Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
    Z = Z.reshape(xx1.shape)
    plt.contourf(xx1, xx2, Z, alpha=0.3, cmap=cmap)
    plt.xlim(xx1.min(), xx1.max())
    plt.ylim(xx2.min(), xx2.max())
```

Answer (Contd.)

Code

Plotting decision surface

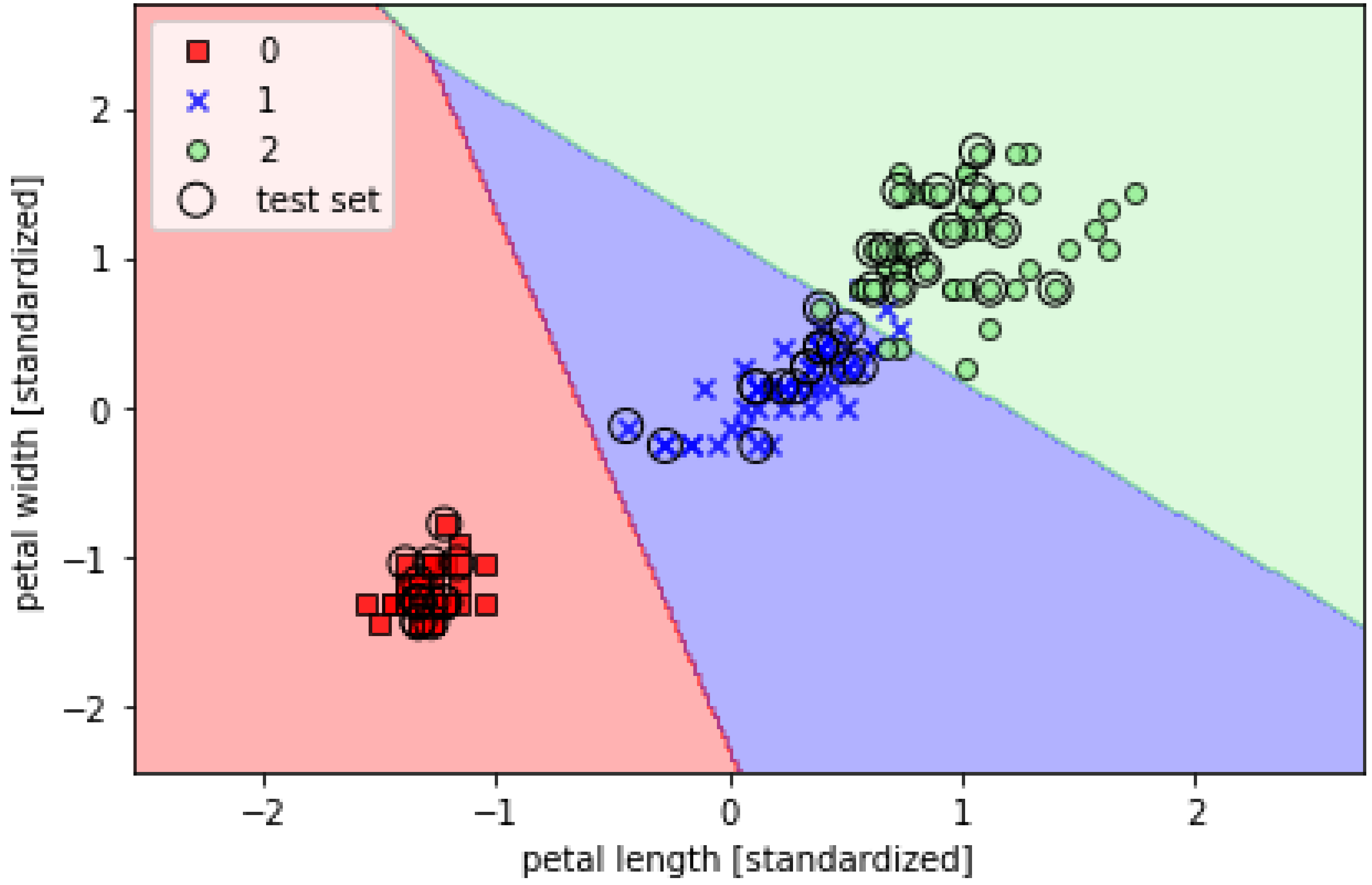
```
for idx, cl in enumerate(np.unique(y)):
        plt.scatter(x=X[y == cl, 0], y=X[y == cl, 1],alpha=0.8,
c=colors[idx],marker=markers[idx], label=cl, edgecolor='black')

    if test_idx:
        X_test, y_test = X[test_idx, :], y[test_idx]
        plt.scatter(X_test[:, 0], X_test[:,
1],c='',edgecolor='black',alpha=1.0,linewidth=1,marker='o',s=100, label='test set')

X_combined_std = np.vstack((X_train_std, X_test_std))
y_combined = np.hstack((y_train, y_test))
```

Answer (Contd.)

Output plot



simplilearn

Answer (Contd.)

Code

Training logistic regression model

```
lr = LogisticRegression(C=100.0, random_state=1)
lr.fit(X_train_std, y_train)
plot_decision_regions(X_combined_std, y_combined,
                      classifier=lr, test_idx=range(105, 150))
plt.xlabel('petal length [standardized]')
plt.ylabel('petal width [standardized]')
plt.legend(loc='upper left')
plt.tight_layout()
#plt.savefig('images/03_06.png', dpi=300)
plt.show()
lr.predict_proba(X_test_std[:3, :])
lr.predict_proba(X_test_std[:3, :]).sum(axis=1)
lr.predict_proba(X_test_std[:3, :]).argmax(axis=1)
lr.predict(X_test_std[:3, :])
lr.predict(X_test_std[0, :].reshape(1, -1))
```

Key Takeaways

Now, you are able to:

- Understand the different types of supervised learning
- Build various regression models

Knowledge Check

Knowledge Check 1	In the equation of a straight line $Y = mX + c$, the term m is the:

a. Slope

b. Independent Variable

c. Dependent Variable

d. Intercept

Knowledge Check 1

In the equation of a straight line Y = mX + c, the term m is the:

a. Slope

b. Independent Variable

c. Dependent Variable

d. Intercept

The correct answer is **a. Slope**

In the equation of a straight line Y = mX + c, m represents the slope, and c is any constant.

Knowledge Check 2

The standard error of the estimate is a measure of:

a. Explained variation

b. Variation around the regression line

c. Variation of the X variable

d. Total variation of the Y variable

Knowledge Check

2

The standard error of the estimate is a measure of:

a. Explained variation

b. Variation around the regression line

c. Variation of the X variable

d. Total variation of the Y variable

The correct answer is **b. Variation around the regression line**

The standard error of the estimate is a measure of the variation around the regression line.

Lesson-End Project

Health Insurance Cost

Duration: 20 mins.

Problem Statement:

Health insurance has become an indispensable part of our lives in recent years and people are paying for them to cover up losses caused either by accident or other unpredicted factors.

You are provided with medical costs dataset that has features such as Age, Cost, BMI

Objective:

- Determine the factors that contribute the most in the calculation of insurance costs
- Predict the health Insurance Cost

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Thank You