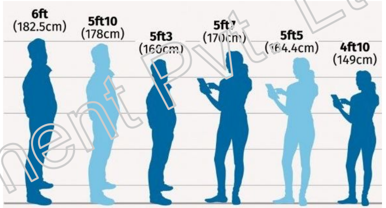
Logistic Regression

Dr. Ram Prasad K
VisionCog R&D

ram.krish@visioncog.com
https://www.visioncog.com

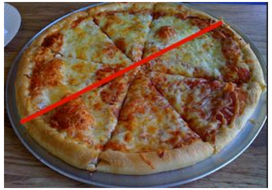
Classification vs Regression



Regression

i.e., **predicting** a continuous value by learning **relationship** between **dependent** and **independent** variables.

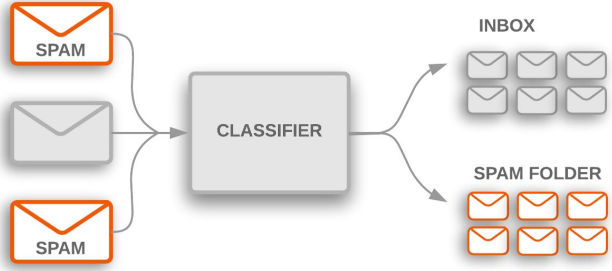
Output is numeric value.



Classification

Identifying which of the **category** a new observation belong.

Output is class label.



VisionCog Research and Development Pvt. Ltd.

Logistic Regression

Logistic Regression estimates **probability** that an instance belongs to a particular **class**.

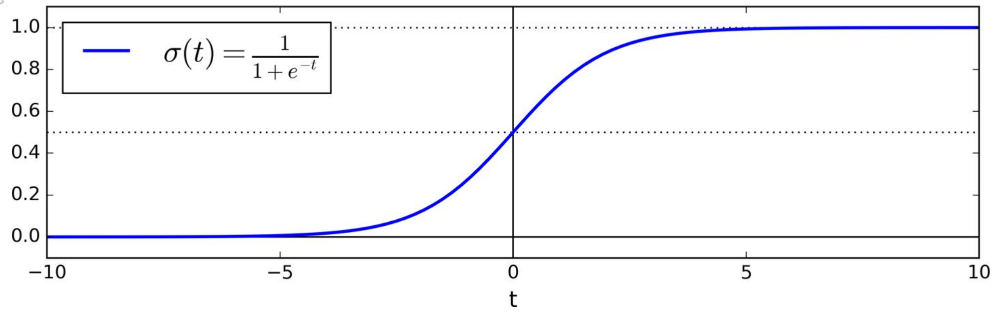
Just like linear regression, logistic regression also finds weighted sum of the inputs but instead of the continuous value, it outputs its sigmoid result.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \longrightarrow$$

Linear Regression (Multivariate)

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$$

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$



VisionCog Research and Development Pvt. Ltd.

Logistic Regression

Logistic Regression model prediction (binary classifier)

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$$

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

No analytical solution

It is a convex function

Gradient descent can be used to solve the problem.

Cost function to optimize

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(\hat{p}^{(i)}) + (1 - y^{(i)})log(1 - \hat{p}^{(i)})]$$

VisionCog Research and Development Pvt. Ltd.

Softmax Regression

Softmax Regression

Softmax regression is logistic regression extended for multiple classes.

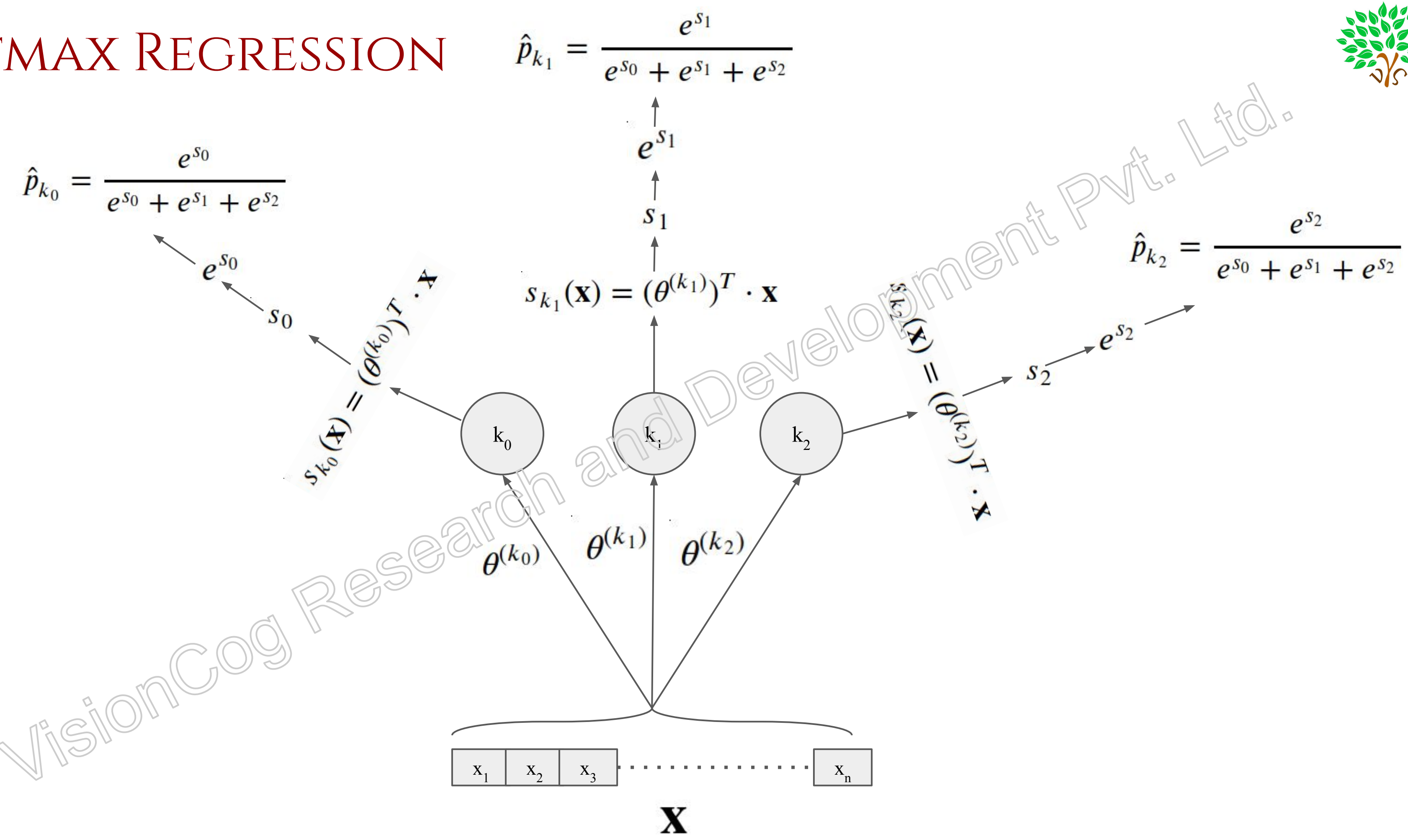
Logistic regression supports only binary classification.

Softmax regression:

- For each class ***k***, a score is estimated.
- Then estimates probability of each class by applying softmax function.
- The predicted class is the one with highest estimated probability.

VisionCog Research and Development Pvt. Ltd.

Softmax Regression



VisionCog Research and Development Pvt. Ltd.

Softmax Regression

Score

$$s_k(\mathbf{x}) = \left(\boldsymbol{\theta}^{(k)}\right)^T \mathbf{x}$$

Probability

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)}$$

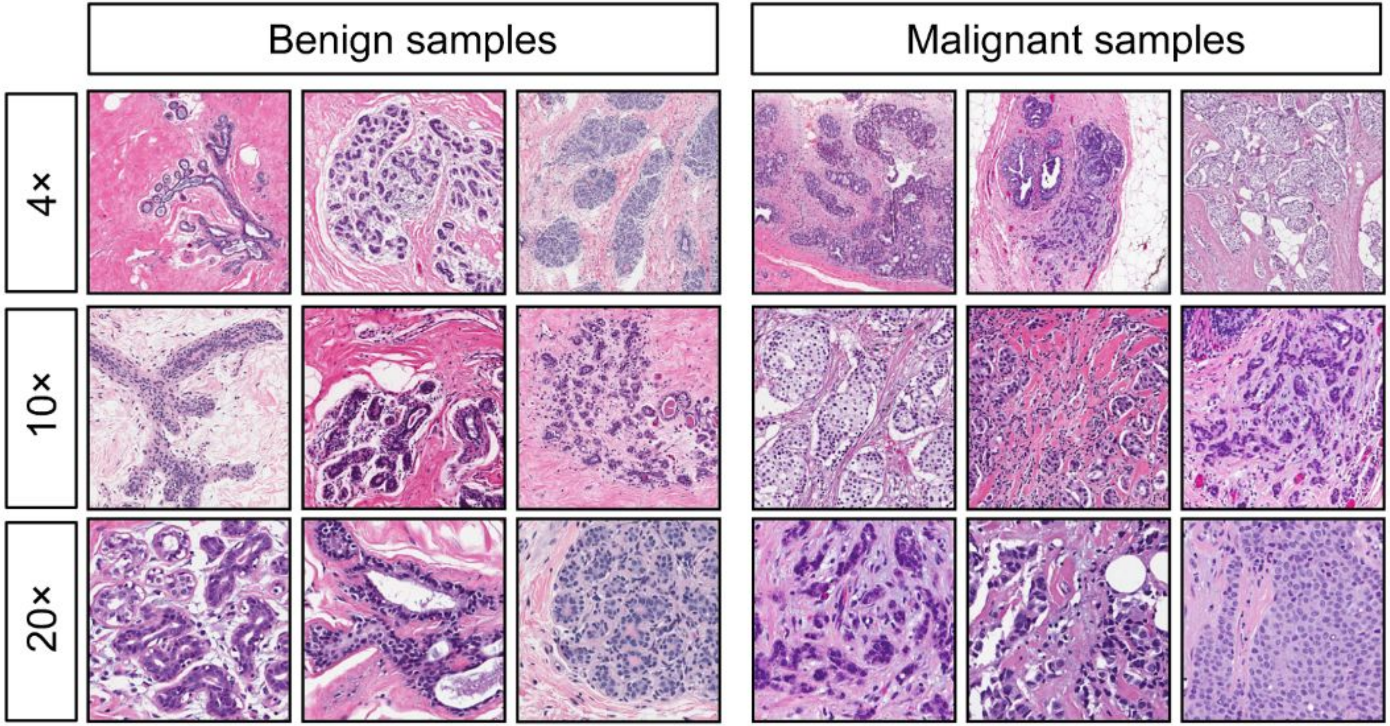
Class prediction

$$\hat{y} = \underset{k}{\operatorname{argmax}}\ \sigma(\mathbf{s}(\mathbf{x}))_k = \underset{k}{\operatorname{argmax}}\ s_k(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\left(\left(\boldsymbol{\theta}^{(k)}\right)^T \mathbf{x}\right)$$

VisionCog Research and Development Pvt. Ltd.

Logistic Regression

Breast cancer wisconsin (diagnostic) dataset



Logistic Regression

```python
import numpy as np
from sklearn import datasets

cancerDB = datasets.load_breast_cancer()

print(cancerDB.DESCR)
```

```
Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the three
        largest values) of these features were computed for each image,
        resulting in 30 features.  For instance, field 3 is Mean Radius, field
        13 is Radius SE, field 23 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign
```

VisionCog Research and Development Pvt. Ltd.

Logistic Regression

:Summary Statistics:

	Min	Max
radius (mean):	6.981	28.11
texture (mean):	9.71	39.28
perimeter (mean):	43.79	188.5
area (mean):	143.5	2501.0
smoothness (mean):	0.053	0.163
compactness (mean):	0.019	0.345
concavity (mean):	0.0	0.427
concave points (mean):	0.0	0.201
symmetry (mean):	0.106	0.304
fractal dimension (mean):	0.05	0.097
radius (standard error):	0.112	2.873
texture (standard error):	0.36	4.885
perimeter (standard error):	0.757	21.98
area (standard error):	6.802	542.2
smoothness (standard error):	0.002	0.031
compactness (standard error):	0.002	0.135
concavity (standard error):	0.0	0.396
concave points (standard error):	0.0	0.053
symmetry (standard error):	0.008	0.079
fractal dimension (standard error):	0.001	0.03
radius (worst):	7.93	36.04
texture (worst):	12.02	49.54
perimeter (worst):	50.41	251.2
area (worst):	185.2	4254.0
smoothness (worst):	0.071	0.223
compactness (worst):	0.027	1.058
concavity (worst):	0.0	1.252
concave points (worst):	0.0	0.291
symmetry (worst):	0.156	0.664
fractal dimension (worst):	0.055	0.208

VisionCog Research and Development Pvt. Ltd.

Logistic Regression

```
X = cancerDB.data   # Loading the features
y = cancerDB.target # Loading the target

print(X.shape)
# (569, 30)

print(y.shape)
# (569,)

print(set(y))
# {0, 1}
```

VisionCog Research and Development Pvt. Ltd.

Logistic Regression

```
from sklearn.model_selection import train_test_split

# Splitting the dataset into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    stratify=y, random_state=42)

print(X_train.shape)
# (455, 30)

print(X_test.shape)
# (114, 30)
```

VisionCog Research and Development Pvt. Ltd.

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

# Creating an object for LogisticRegression class
model_LR = LogisticRegression()

# Training the model to estimate the parameters
model_LR.fit(X_train, y_train)

# Evaluate the accuracy of the model using test set
accuracy = model_LR.score(X_test, y_test)

print(accuracy)
# 0.96491
```

VisionCog Research and Development Pvt. Ltd.