## **LAB 11**

**Logistic Regression**

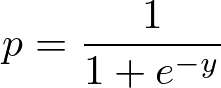
Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence. It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

**Linear Regression Equation:**

eq1

Where, y is a dependent variable and x1, x2 ... and Xn are explanatory variables.

**Sigmoid Function:**



**Apply Sigmoid function on linear regression:**



**Properties of Logistic Regression:**

* The dependent variable in logistic regression follows Bernoulli Distribution.
* Estimation is done through maximum likelihood.
* No R Square, Model fitness is calculated through Concordance, KS-Statistics.

**Logistic Regression Model building**

Let's first load the required Pima Indian Diabetes dataset using the pandas' read CSV function. You can download data from the following link: [**https://www.kaggle.com/uciml/pima-indians-diabetes-database**](https://www.kaggle.com/uciml/pima-indians-diabetes-database) or select a dataset from DataCamp: [**https://www.datacamp.com/workspace/datasets**](https://www.datacamp.com/workspace/datasets).

**Importing Required Libraries**

Let's first load the required libraries.

# Load libraries

import pandas as pd

from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

**Loading Data**

Let's first load the required Pima Indian Diabetes dataset using pandas' read CSV function.

col\_names =['Pregnant', 'Glucose', 'Bp', 'Skin', 'Insulin', 'Bmi', 'Pedigree', 'Age', 'Label']

# load dataset

pima = pd.read\_csv("https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.csv", header=None, names=col\_names)

pima.head()

**Feature Selection**

Here, you need to divide given columns into two types of variables dependent(or target variable) and independent variable(or feature variables).

#split dataset in features and target variable

feature\_cols = ['Pregnant', 'Insulin', 'Bmi', 'Age','Glucose','Bp','Pedigree']

X = pima[feature\_cols] # Features

y = pima.label # Target variable

**Splitting Data**

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split the dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size.

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1) # 70% training and 30% test

**Model Development and Prediction**

First, import the Logistic Regression module and create a Logistic Regression classifier object using the LogisticRegression() function with random\_state for reproducibility.

Then, fit your model on the train set using fit() and perform prediction on the test set using predict().

# instantiate the model (using the default parameters)

logreg = LogisticRegression(random\_state=16)

# fit the model with data

logreg.fit(X\_train, y\_train)

y\_pred = logreg.predict(X\_test)

**Evaluating Model**

Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.

# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

Well, you got a classification rate of 78.35%, considered as good accuracy.

##### **Lab Tasks**

# **Task # 1:** Using python implement Logistic Regression Algorithm on Heart Attack Analysis & prediction dataset to predict the chances of heart failure in a person.