Department of Information Engineering Technology

The Superior University, Lahore

Artificial Intelligence Lab

Experiment No.4

Supervised Learning: Logistic Regression

Prepared for

_____

By:

Name: _____

ID: _____

Section: _____

Semester: _____

Total Marks: _____

Obtained Marks: _____

Signature: _____

Date: _____

Experiment No. 4

Supervised Learning: Logistic Regression

(Rubrics)

Name: _____ **Roll No. : _____**

A. PSYCHOMOTOR

Sr. No.	Criteria	Allocated Marks	Unacceptable 0%	Poor 25%	Fair 50%	Good 75%	Excellent 100%	Total Obtained
1	Follow Procedures	1	0	0.25	0.5	0.75	1	
2	Software and Simulations	2	0	0.5	1	1.5	2	
3	Accuracy in Output Results	3	0	0.75	1.5	2.25	3	
Sub Total		**6**	**Sub Total marks Obtained in Psychomotor(P)**					

B. AFFECTIVE

Sr. No.	Criteria	Allocated Marks	Unacceptable 0%	Poor 25%	Fair 50%	Good 75%	Excellent 100%	Total Obtained
1	Respond to Questions	1	0	0.25	0.5	0.75	1	
2	Lab Report	1	0	0.25	0.5	0.75	1	
3	Assigned Task	2	0	0.5	1	1.5	2	
Sub Total		**4**	**Sub Total marks Obtained in Affective (A)**					

Instructor Name:_____ **Total Marks (P+A): _____**

Instructor Signature:_____ **Obtained Marks (P+A): _____**

Introduction to Linear Regression

Linear regression is a fundamental statistical and machine learning technique used for modeling and analyzing the relationship between a dependent variable and one or more independent variables. It is a type of supervised learning algorithm, often applied in predictive modeling and data analysis tasks.

Here are the key components and characteristics of linear regression:

Dependent Variable (Target): Linear regression aims to predict or explain a dependent variable (also known as the target or response variable). This is the variable you want to understand or make predictions about. In mathematical notation, the dependent variable is typically denoted as "Y."

Independent Variables (Predictors or Features): Linear regression considers one or more independent variables (also known as predictors or features) that are believed to influence or explain variations in the dependent variable. In mathematical notation, the independent variables are typically denoted as "X."

Linear Relationship: Linear regression assumes that there is a linear relationship between the independent variables and the dependent variable. In simple linear regression (with one independent variable), this relationship is represented by a straight line. In multiple linear regression (with multiple independent variables), it's represented by a hyperplane.

Linear Equation: The linear relationship is expressed as a linear equation, typically in the form of:

$$y = mx + c$$

- y is the dependent variable.
- x is the independent variables.
- m are the coefficients to be estimated or slope of the line.
- c is the y-intercept

Linear Regression with Python

Linear regression is a simple yet powerful machine learning technique used for modeling the relationship between a dependent variable (target) and one or more independent variables (features). In Python, you can perform linear regression using various libraries, but one of the most commonly used libraries is scikit-learn.

1. **Importing Libraries:** Start by importing the necessary libraries.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

2. **Load Data:** Load your dataset using pandas or any other method suitable for your data source.

```
df = pd.read_csv("carprices.csv")
df.head()
```

3. **Scatter Plot:** You can visualize by plotting the actual vs. predicted values or any other relevant visualizations.

```
plt.scatter(df['Mileage'],df['Sell Price($)'])
plt.scatter(df['Age(yrs)'],df['Sell Price($)'])
```

4. **Prepare Data:** Organize your data into the dependent variable (target) and independent variables (features).

```
X = df[['Mileage','Age(yrs)']]
y = df['Sell Price($)']
```

5. **Split the dataset into Training and Testing sets:** Divide your dataset into two parts: a training set used to train the model and a testing set used to evaluate its performance. The *train_test_split* function helps you do this.

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,
random_state=10)
```

6. **Create and Train the Model:** Create a linear regression model and fit it to your training data.

```
clf = LinearRegression()
clf.fit(X_train, y_train)
```

7. **Make Predictions:** Use the trained model to make predictions on the test data.

```
clf.predict(X_test)
```

8. **Check Accuracy:** *clf.score(X_test, y_test*) is typically used to calculate the accuracy of a classification model, not for linear regression. In scikit-learn, the score method is used to compute the accuracy of classifiers, and it expects the features (X_test) and the true labels (y_test) as input.

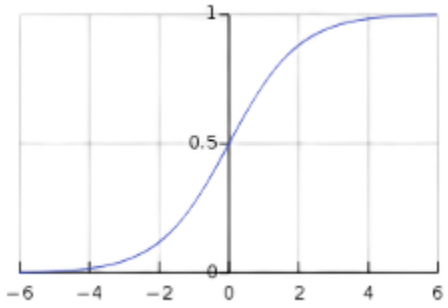
```
clf.score(X_test, y_test)
```

. Introduction to Logistic Regression

Logistic Regression is a fundamental and widely-used statistical method in machine learning and statistics for binary classification tasks. It's named "logistic regression" because it's based on the logistic function, also known as the sigmoid function, which is used to model the probability of a binary outcome.

Here's an introduction to logistic regression:

- **Binary Classification:** Logistic regression is primarily used for binary classification, where the goal is to predict one of two possible outcomes based on one or more independent variables (features). For example, it can be used to predict whether an email is spam (1) or not spam (0), whether a patient has a disease (1) or doesn't have it (0), etc.
- **Multiclass Classification:** Multiclass classification is an extension of binary classification, where the goal is to categorize data points into one of three or more classes or categories. In multiclass classification, each data point belongs to exactly one class out of several possible classes. It's a common machine learning task used in various applications such as image classification, natural language processing, and medical diagnosis, among others.
- **Sigmoid Function:** The logistic function, or sigmoid function, has an S-shaped curve and is defined as:



$$\textit{Equation: } \sigma(x) = \frac{1}{1+e^{-x}}$$

The sigmoid function maps any real-valued number to the range [0, 1]. This makes it suitable for modeling probabilities.

- **Model Parameters:** Logistic regression has model parameters that are learned during the training process. These parameters include weights (coefficients) for each feature and an intercept term. The logistic regression model computes a linear combination of the input features and applies the sigmoid function to it.
- **Maximum Likelihood Estimation:** During training, logistic regression uses the principle of maximum likelihood estimation to adjust the model parameters. The goal is to find the parameter values that maximize the likelihood of the observed data, given the model.

In summary, logistic regression is a foundational technique for binary classification tasks, offering a probabilistic approach to estimating class probabilities. It's widely used in various fields, including healthcare, finance, marketing, and more, due to its simplicity, interpretability, and effectiveness in solving binary classification problems.

Logistic Regression with Python

Logistic Regression is a statistical method used for binary classification, where the goal is to predict one of two possible outcomes (usually labeled as 0 and 1) based on one or more independent variables (features). Despite its name, logistic regression is a classification algorithm, not a regression algorithm. It's widely used in machine learning and statistics for tasks like spam detection, medical diagnosis, and more.

Here's a step-by-step guide on how to perform logistic regression in Python using the scikit-learn library:

1. **Importing Libraries:** Start by importing the necessary libraries.

```
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import math
```

2. **Load Data:** Load your dataset using pandas or any other method suitable for your data source.

```
df = pd.read_csv("insurance_data.csv")
df.head()
```

3. **Scatter Plot:** You can visualize by plotting the actual vs. predicted values or any other relevant visualizations.

```
plt.scatter(df.age,df.bought_insurance,marker='+',color='red')
```

4. **Prepare Data:** Organize your data into the dependent variable (target) and independent variables (features).

```
x=df['age']
y=df['bought_insurance']
```

5. **Split the dataset into Training and Testing sets:** Divide your dataset into two parts: a training set used to train the model and a testing set used to evaluate its performance. The *train_test_split* function helps you do this.

```
X_train, X_test, y_train, y_test = train_test_split(x,y,train_size=0.8)
```

6. **Create and Train the Model:** Create a logistic regression model and fit it to your training data.

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

7. **Make Predictions:** Use the trained model to make predictions on the test data.

```
y_predicted = model.predict(X_test)
```

8. **Probability Predictions:** The *model.predict_proba(X_test*) method is used in machine learning, particularly for classification tasks, to obtain the probability estimates of the different classes for each data point in the test set X_test. This method is commonly associated with models like logistic regression, support vector machines with a probability output, and some types of decision trees, among others.

```
model.predict_proba(X_test)
```

9. **Check Accuracy:** *clf.score(X_test, y_test*) is typically used to calculate the accuracy of a classification model. In scikit-learn, the score method is used to compute the accuracy of classifiers, and it expects the features (X_test) and the true labels (y_test) as input.

```
model.score(X_test,y_test)
```

Calculation of Sigmoid Function manually:

```
model.coef_
model.intercept_
#Lets defined sigmoid function now and do the math with hand
def sigmoid(x):
  return 1 / (1 + math.exp(-x))
def prediction_function(age):
    z = 0.042 * age - 1.53 # 0.04150133 ~ 0.042 and -1.52726963 ~ -1.53
    y = sigmoid(z)
    return y
age = 35
prediction_function(age)
age = 43
prediction_function(age)
```

Confusion Matrix:

A confusion matrix is a common tool for evaluating the performance of a classification model. It provides a detailed summary of how many true positives, true negatives, false positives, and false negatives your model produced. In Python, you can create a confusion matrix using libraries like scikit-learn. Here's how to do it:

```
y_predicted = model.predict(X_test)

cm = confusion_matrix(y_test, y_predicted)
cm
# Plotting Graph of Confusion Matrix
plt.figure(figsize = (10,7))
```

```
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Lab task:

1. Download employee retention dataset from here: https://www.kaggle.com/giripujar/hr-analytics.

   - Now do some exploratory data analysis to figure out which variables have direct and clear impact on employee retention (i.e. whether they leave the company or continue to work)
   - Plot bar charts showing impact of employee salaries on retention
   - Plot bar charts showing corelation between department and employee retention
   - Now build logistic regression model using variables that were narrowed down in step 1
   - Measure the accuracy of the model

2. Use sklearn.datasets iris flower dataset to train your model using logistic regression. You need to figure out accuracy of your model and use that to predict different samples in your test dataset. In iris dataset there are 150 samples containing following features,
   - Sepal Length
   - Sepal Width
   - Petal Length
   - Petal Width

   Using above 4 features you will classify a flower in one of the three categories,

   - Setosa
   - Versicolour
   - Virginica

3. Download any weather dataset from the sklearn and perform the train_test_split method . also check the accuracy of your model.

Home Task:

1. Use the Sklearn dataset of Breast Cancer Wisconsin dataset. Explore the dataset by examining its structure, features, and target variable. Evaluate the model's performance using the following metrics:
   - Accuracy
   - Precision

   Make predictions on new, unseen data using the trained logistic regression model (Binary).

2. Develop a machine learning model that can classify emails into multiple categories based on their content. This problem involves multiclass classification using logistic regression. Email dataset with labeled categories (e.g., "Spam," "Promotions," "Personal," "Work," etc.). You can use a publicly available dataset or create a synthetic one for educational purposes. Create an instance of the Logistic Regression class from scikit-learn for multiclass classification.