WEEK -05: LOGISTIC REGRESSION AND STOCHASTIC GRADIENT DESCENT (SGD)

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class or not. It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables. It is a powerful tool for decision-making. For example email spam or not.

It is used for classification algorithms its name is logistic regression. it's referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class. The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.

It is used for predicting the categorical dependent variable using a given set of independent variables.

- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1. o The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form.
- The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Terminologies involved in Logistic Regression:

Here are some common terms involved in logistic regression:

- **Independent variables:** The input characteristics or predictor factors applied to the dependent variable's predictions.
- **Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
- **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.
- **Odds:** It is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.
- **Log-odds:** The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.

- **Coefficient:** The logistic regression model's estimated parameters, show how the independent and dependent variables relate to one another.
- **Intercept:** A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.
- **Maximum likelihood estimation:** The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

How does Logistic Regression work?

The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.

Let the independent input features be

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ x_{21} & \dots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}$$

and the dependent variable is Y having only binary value i.e. 0 or 1.

$$Y = \begin{cases} 0 & \text{if } Class\ 1 \\ 1 & \text{if } Class\ 2 \end{cases}$$

then apply the multi-linear function to the input variables X

$$z = (\sum_{i=1}^{n} w_i x_i) + b$$

x_i $w_i = [w_1, w_2, w_3, \cdots, w_m]$

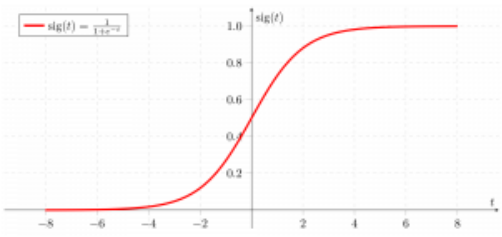
Here is the ith observation of X, is the weights or Coefficient, and b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$z = w \cdot X + b$$

Sigmoid Function

Now we use the sigmoid function where the input will be z and we find the probability between 0 and 1. i.e predicted y.

$$\sigma(z) = \frac{1}{1-e^{z}}$$



Sigmoid function

As shown above, the figure sigmoid function converts the continuous variable data into the probability i.e. between 0 and 1.

- $\sigma(z)$ tends towards 1 as $z \to \infty$
- $\sigma(z)$ tends towards 0 as $z \to -\infty$
- is always bounded between 0 and 1

where the probability of being a class can be measured as:

$$P(y=1) = \sigma(z)$$
$$P(y=0) = 1 - \sigma(z)$$

Logistic Regression Equation

The odd is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur. so odd will be

$$\frac{p(x)}{1-p(x)} = e^z$$

Applying natural log on odd. then log odd will be

$$\log\left[\frac{p(x)}{1-p(x)}\right] = z$$
$$\log\left[\frac{p(x)}{1-p(x)}\right] = w \cdot X + b$$

then the final logistic regression equation will be:

$$p(X; b, w) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} = \frac{1}{1 + e^{-w \cdot X + b}}$$

Likelihood function for Logistic Regression

The predicted probabilities will p(X;b,w) = p(x) for y=1 and for y = 0 predicted probabilities will 1-p(X;b,w) = 1-p(x)

$$L(b, w) = \prod_{i=1} np(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

Taking natural logs on both sides

$$\begin{aligned} l(b, w) = \log(L(b, w)) &= \sum_{i=1}^{n} y_i \log p(x_i) \ + \ (1 - y_i) \log(1 - p(x_i)) \\ &= \sum_{i=1}^{n} y_i \log p(x_i) + \log(1 - p(x_i)) - y_i \log(1 - p(x_i)) \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log \frac{p(x_i)}{1 - p(x_i} \\ &= \sum_{i=1}^{n} - \log 1 - e^{-(w \cdot x_i + b)} + \sum_{i=1}^{n} y_i (w \cdot x_i + b) \\ &= \sum_{i=1}^{n} - \log 1 + e^{w \cdot x_i + b} + \sum_{i=1}^{n} y_i (w \cdot x_i + b) \end{aligned}$$

Gradient of the log-likelihood function

To find the maximum likelihood estimates, we differentiate w.r.t w,

$$\frac{\partial J(l(b,w)}{\partial w_j} = -\sum_{i=n}^{n} \frac{1}{1+e^{w\cdot x_i+b}} e^{w\cdot x_i+b} x_{ij} + \sum_{i=1}^{n} y_i x_{ij}$$

$$= -\sum_{i=n}^{n} p(x_i;b,w) x_{ij} + \sum_{i=1}^{n} y_i x_{ij}$$

$$= \sum_{i=n}^{n} (y_i - p(x_i;b,w)) x_{ij}$$

Assumptions for Logistic Regression

The assumptions for Logistic regression are as follows:

- **Independent observations:** Each observation is independent of the other. meaning there is no correlation between any input variables.
- **Binary dependent variables:** It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories softmax functions are used.
- **Linearity relationship between independent variables and log odds:** The relationship between the independent variables and the log odds of the dependent variable should be linear.
- **No outliers:** There should be no outliers in the dataset.
- **Large sample size:** The sample size is sufficiently large

Logistic regression is a commonly used algorithm for binary classification problems. Here's a Python program for logistic regression with an example using the popular Iris dataset for classification:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix

# Load the Iris dataset
iris = datasets.load_iris()
X = iris.data[:, :2]  # We'll use only the first two features for simplicity
y = (iris.target != 0) * 1  # Convert target labels to binary (0 or 1)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardize the feature data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create a logistic regression model
model = LogisticRegression(solver='liblinear')

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
print("Confusion Matrix:")
```

```
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Plot the decision boundary
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.01), np.arange(y_min, y_max, 0.01))
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, cmap=plt.cm.RdBu, alpha=0.8)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.RdBu)
plt.xlabel('Sepal Length (standardized)')
plt.ylabel('Sepal Width (standardized)')
plt.title('Logistic Regression Decision Boundary')
plt.show()
```

In this program:

1. We load the Iris dataset and select only the first two features for binary classification.
2. We split the data into training and testing sets using **train_test_split**.
3. We standardize the feature data using **StandardScaler** to have zero mean and unit variance.
4. We create a logistic regression model using **LogisticRegression** from scikit-learn.
5. We train the model on the training data.
6. We make predictions on the test data and evaluate the model's performance using a confusion matrix and a classification report.
7. Finally, we plot the decision boundary of the logistic regression model to visualize how it separates the two classes.

You can adjust the dataset, features, and model parameters as needed for your specific classification problem.

Stochastic gradient descent (SGD)

Ref: https://machinelearningmastery.com/linear-regression-tutorial-using-gradient-descent-for-machine-learning/

Questions

Note: "Refer to the table that contains the average annual gold rate from 1965 to 2022 and the year-wise silver prices available at Week-4 ML lab manual."

1. Let's say we have a fictional dataset of pairs of variables, a mother and her daughter's heights:

mother height	daughter height
58	60
62	60
60	58
64	60
67	70
70	72

height of mother(x)/daughter (y) pairs

Create a CSV file for the above training data and write a Python function program to find the fitted linear regression with gradient descent technique. Compare the coefficients obtained from the sklearn model with your program. Compute the error, MSE and RMSE. Plot the graph Daughter height (Y-axis) vs Mother height (X-axis) with blue colour. Also, plot the line of best fit with red colour. Predict her daughter's height with given a new mother height as 63. Plot the graph of error in y-axis and iteration in x-axis with 4 epochs (6x4=24 iterations).

2.

Hours of Study (X)	Pass (Y)
1	0
2	0
3	0
4	0
5	1
6	1
7	1
8	1

Here, X is the number of hours of study, and Y is the outcome (0 for fail, 1 for pass).

Create a CSV file for the above training data and write a Python function program to find the fitted logistic regression with gradient descent technique. Compare the coefficients obtained from the sklearn model with your program. Compute the predicted y and assign the class label (prediction = 0 IF p(fail) < 0.5 and prediction = 1 IF p(pass) >= 0.5) and compute the accuracy. Find the error for each iteration and predict the probability that a student will pass the exam if they study for a) 3.5 hours b) 7.5 hours. Plot the graph of error in y-axis and iteration in x-axis with 3 epochs (8x3=24 iterations).

3.

	x_1	x_2	y
1)	4	1	2
2)	2	8	-14
3)	1	0	1
4)	3	2	-1
5)	1	4	-7
6)	6	7	-8

Consider the above dataset with two independent variables (X1 and X2) and a dependent variable (Y). Implement in python, how you can perform the logistic regression to model the relationship between the independent variables and the dependent variable.

Additional Questions

1. Write a python program for SGD by considering the year wise gold and silver price data. Compare the coefficients obtained from sklearn model with your program. Compute the error, MSE and RMSE. Predict the gold price with the year 2025 for 1 gram and, gold and silver price with the year 2024 for 1 gram.

2. Suppose you have a gold/silver price dataset with a single independent variable (X) and a dependent variable (Y). You want to fit a logistic regression model to this data. Develop an example code snippet in Python.

3. Consider the gold and/or silver price dataset and to evaluate a logistic regression model using ROC and AUC:

1. Calculate predicted probabilities for each instance in the test set.
2. Plot the ROC curve using the True Positive Rate (Sensitivity) and False Positive Rate.
3. Calculate the AUC to summarize the model's performance.