ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DAY – 14

Date: Jul 10, 2025

Logistic Regression

Logistic Regression is a supervised machine learning algorithm used for classification problems. Unlike linear regression which predicts continuous values it predicts the probability that an input belongs to a specific class. It is used for binary classification where the output can be one of two possible categories such as Yes/No, True/False or 0/1. It uses sigmoid function to convert inputs into a probability value between 0 and 1. In this article, we will see the basics of logistic regression and its core concepts.

Types of Logistic Regression

Logistic regression can be classified into three main types based on the nature of the dependent variable:

1. **Binomial Logistic Regression:** This type is used when the dependent variable has only two possible categories. Examples include Yes/No, Pass/Fail or 0/1. It is the most common form of logistic regression and is used for binary classification problems.
2. **Multinomial Logistic Regression:** This is used when the dependent variable has three or more possible categories that are not ordered. For example, classifying animals into categories like "cat," "dog" or "sheep." It extends the binary logistic regression to handle multiple classes.
3. **Ordinal Logistic Regression:** This type applies when the dependent variable has three or more categories with a natural order or ranking. Examples include ratings like "low," "medium" and "high." It takes the order of the categories into account when modeling.

Implementation for Logistic Regression

1. Binomial Logistic regression:

In binomial logistic regression, the target variable can only have two possible values such as "0" or "1", "pass" or "fail". The sigmoid function is used for prediction.

```
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X, y = load_breast_cancer(return_X_y=True)


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=23)
```

```
clf = LogisticRegression(max_iter=10000, random_state=0)
clf.fit(X_train, y_train)

acc = accuracy_score(y_test, clf.predict(X_test)) * 100
print(f"Logistic Regression model accuracy: {acc:.2f}%")
```

Output:

```
Logistic Regression model accuracy (in %): 96.49%
```

2. Multinomial Logistic Regression:

Target variable can have 3 or more possible types which are not ordered i.e types have no quantitative significance like “disease A” vs “disease B” vs “disease C”.

In this case, the softmax function is used in place of the sigmoid function. Softmax function for K classes will be:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Here KK represents the number of elements in the vector zz and i,ji,j iterates over all the elements in the vector.

Then the probability for class cc will be:

$$P(Y = c|\overrightarrow{X} = x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{k=1}^{K} e^{w_k \cdot x + b_k}}$$

Below is an example of implementing multinomial logistic regression using the Digits dataset from scikit-learn:

```
from sklearn.model_selection import train_test_split
from sklearn import datasets, linear_model, metrics

digits = datasets.load_digits()

X = digits.data

y = digits.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)
```

```
reg = linear_model.LogisticRegression(max_iter=10000, random_state=0)
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)

print(f"Logistic Regression model accuracy: {metrics.accuracy_score(y_test, y_pred) *
100:.2f}%")
```

Output:

```
Logistic Regression model accuracy: 96.66%
```