Machine Learning Workflow: Linear and Logistic Regression

1. Linear Regression Workflow

Step 1: Data Reading

First, the dataset is imported into the working environment. The data should include the independent variable (X) and the dependent variable (Y).

For example, for predicting weight based on height, we can load the data as:

```
X = np.array([150, 160, 170, 180, 190]).reshape(-1, 1)  # Height in cm
Y = np.array([55, 65, 70, 80, 90])  # Weight in kg
```

Step 2: Model Selection

We select a linear regression model. In Python, we can use the LinearRegression class from sklearn.

```
model = LinearRegression()
model.fit(X, Y)
```

Step 3: Coefficient Determination

After fitting the model, we can retrieve the coefficient (slope) and intercept:

```
slope = model.coef_[0]
intercept = model.intercept_
```

Step 4: Prediction

To predict the dependent variable for a new independent variable, use the model to make predictions:

```
X_new = np.array([[175]])  # New height value
y_pred = model.predict(X_new)
```

2. Logistic Regression Workflow

Step 1: Data Reading

For logistic regression, the dataset consists of an independent variable (X) and a binary dependent variable (y).

For example, predicting if a person likes a movie based on rating:

```
X = np.array([[2], [3], [4], [5], [6], [7], [8], [9]])
y = np.array([0, 0, 0, 0, 1, 1, 1, 1])
```

Step 2: Model Selection

We choose a logistic regression model using the LogisticRegression class from sklearn.

```
log_model = LogisticRegression()
log_model.fit(X, y)
```

Step 3: Coefficient Determination

The logistic regression model provides the coefficient (slope) and intercept:

```
coef = log_model.coef_[0]
intercept = log_model.intercept_
```

Step 4: Prediction

To make predictions and get probabilities for a new value, use:

```
X_new = np.array([[7]])  # New rating value
y_pred = log_model.predict(X_new)
y_prob = log_model.predict_proba(X_new)
```