

# Title of Project

## Hill and Valley Classification Using Logistic Regression

---

### • Objective

To classify points in a dataset as either "Hill" or "Valley" using logistic regression. This classification will help in understanding and predicting the nature of the terrain based on given features.

### • Data Source

The dataset used for this project is the Hill Valley Dataset provided by YBI Foundation.

### • Import Library

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

### • Import Data

```
df = pd.read_csv('https://github.com/YBI-
Foundation/Dataset/raw/main/Hill%20Valley%20Dataset.csv')
```

### • Describe Data

```
print(df.head())
print(df.describe())
print(df.columns.tolist())
print(df.shape)
print(df['Class'].value_counts())
print(df.groupby('Class').mean())
```

### • Data Visualization

Visualize some examples of the data points to understand the nature of "Hill" and "Valley":

```
plt.plot(df.iloc[0, :-1])  
plt.title('Valley')  
plt.show()
```

```
plt.plot(df.iloc[1, :-1])  
plt.title('Hill')  
plt.show()
```

- **Data Preprocessing**

Standardize the feature variables to have mean 0 and variance 1:

```
from sklearn.preprocessing import StandardScaler
```

```
ss = StandardScaler()  
X = df.drop(['Class'], axis=1)  
X = ss.fit_transform(X)  
y = df['Class']
```

- **Define Target Variable (y) and Feature Variables (X)**

```
y = df['Class']  
X = df.drop(['Class'], axis=1)
```

- **Train Test Split**

Split the data into training and testing sets:

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y,  
random_state=2529)
```

- **Modeling**

Train a logistic regression model on the training data:

```
from sklearn.linear_model import LogisticRegression
```

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

- **Model Evaluation**

Evaluate the model using confusion matrix and classification report:

```
y_pred = lr.predict(X_test)  
print(confusion_matrix(y_test, y_pred))  
print(classification_report(y_test, y_pred))
```

- **Prediction**

Make predictions on new data:

```
X_new = df.sample(1).drop('Class', axis=1)  
X_new = ss.transform(X_new)  
y_pred_new = lr.predict(X_new)  
print(y_pred_new)  
print(lr.predict_proba(X_new))
```

- **Explanation**

This project aims to classify data points as either "Hill" or "Valley" based on their features using logistic regression. Logistic regression is a statistical model that uses a logistic function to model a binary dependent variable. In this project, the features of the dataset are standardized, and a logistic regression model is trained to predict the class labels. The model's performance is evaluated using confusion matrix and classification report, which provide metrics such as precision, recall, and F1-score. Finally, the trained model is used to make predictions on new data points.