

Hill and valley prediction using Logistic Regression.

Importing Pandas library,scikit library

```
In [35]: import pandas as pd
df=pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/Hill%20Valley%20Dataset.csv")
display(df.head())
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V92	V93	V94	V95	V96	V97
0	39.02	36.49	38.20	38.85	39.38	39.74	37.02	39.53	38.81	38.79	...	36.62	36.92	38.80	38.52	38.07	36.73
1	1.83	1.71	1.77	1.77	1.68	1.78	1.80	1.70	1.75	1.78	...	1.80	1.79	1.77	1.74	1.74	1.80
2	68177.69	66138.42	72981.88	74304.33	67549.66	69367.34	69169.41	73268.61	74465.84	72503.37	...	73438.88	71053.35	71112.62	74916.48	72571.58	66348.97
3	44889.06	39191.86	40728.46	38576.36	45876.06	47034.00	46611.43	37668.32	40980.89	38466.15	...	42625.67	40684.20	46960.73	44546.80	45410.53	47139.44
4	5.70	5.40	5.28	5.38	5.27	5.61	6.00	5.38	5.34	5.87	...	5.17	5.67	5.60	5.94	5.73	5.22

5 rows × 101 columns

```
In [36]: df.isnull().sum()
```

```
Out[36]: V1      0
V2      0
V3      0
V4      0
V5      0
..
V97     0
V98     0
V99     0
V100    0
Class    0
Length: 101, dtype: int64
```

```
In [37]: df.describe()
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...
count	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	1212.000000	...
mean	8169.091881	8144.306262	8192.653738	8176.868738	8128.297211	8173.030008	8188.582748	8183.641543	8154.670066	8120.767574	...
std	17974.950461	17881.049734	18087.938901	17991.903982	17846.757963	17927.114105	18029.562695	18048.582159	17982.390713	17900.798206	...
min	0.920000	0.900000	0.850000	0.890000	0.880000	0.860000	0.870000	0.650000	0.650000	0.620000	...
25%	19.602500	19.595000	18.925000	19.277500	19.210000	19.582500	18.690000	19.062500	19.532500	19.285000	...
50%	301.425000	295.205000	297.260000	299.720000	295.115000	294.380000	295.935000	290.850000	294.565000	295.160000	...
75%	5358.795000	5417.847500	5393.367500	5388.482500	5321.987500	5328.040000	5443.977500	5283.655000	5378.180000	5319.097500	...
max	117807.870000	108896.480000	119031.350000	110212.590000	113000.470000	116848.390000	115609.240000	118522.320000	112895.900000	117798.300000	...

8 rows × 101 columns

Defining Target variable(y) and feature variable(x)

Train Test Splitting of the data

```
In [38]: from sklearn.model_selection import train_test_split
X=df.drop('Class',axis=1)
y=df['Class']
print(X.shape)
print(y.shape)
print(y)
X_train,X_test,y_train,y_test=train_test_split(X,y)
```

```
(1212, 100)
(1212,)
0      0
1      1
2      1
3      0
4      0
..
1207   1
1208   0
1209   1
1210   1
1211   0
Name: Class, Length: 1212, dtype: int64
```

```
In [39]: print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(909, 100)
(303, 100)
(909,)
(303,)
```

Scaling the data

```
In [47]: from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
print(X_train.shape)
X_test=sc.transform(X_test)
```

(909, 100)

Modelling

```
In [42]: from sklearn.linear_model import LogisticRegression
lg=LogisticRegression(random_state=0)
lg.fit(X_train,y_train)
y_pred=lg.predict(X_test)
print(y_pred.shape)
```

(303,)

Accuracy Score

```
In [43]: from sklearn.metrics import accuracy_score,confusion_matrix
print(accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

0.7524752475247525

```
[[151  2]
 [ 73 77]]
```

```
In [ ]: #This project is based on the Dasa science using Python Programming language.
#The main purpose of creating this project is to develop a predictive model using logistic regression that can accurately classify geographical locations as
#either hills or valleys based on a set of input features.
#The model will be trained on a labeled dataset of geographical features and their corresponding classifications,
#and then evaluated on a separate test dataset to measure its performance.
#The ultimate goal of this project is to provide a useful tool for identifying hills and valleys in various geographic locations,
#which can have important applications in fields such as geology, agriculture, and urban planning.
```

```
In [ ]:
```