

Practical 2

Aim: Write a Program to implement regularization to prevent the model from overfitting.

L1 Regularization:

```
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn.linear_model import Lasso

df_train = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')

df_train = df_train.dropna()
df_test = df_test.dropna()

x_train = df_train['x']
x_train = x_train.values.reshape(-1,1)
y_train = df_train['y']
y_train = y_train.values.reshape(-1,1)
x_test = df_test['x']
x_test = x_test.values.reshape(-1,1)
y_test = df_test['y']
y_test = y_test.values.reshape(-1,1)
lasso = Lasso()
lasso.fit(x_train, y_train)
print("Lasso Train RMSE:",
np.round(np.sqrt(metrics.mean_squared_error(y_train,
lasso.predict(x_train))), 50))
print("Lasso Test RMSE:",
np.round(np.sqrt(metrics.mean_squared_error(y_test,
lasso.predict(x_test))), 50))
```

Output:

```
Lasso Train RMSE: 2.8051643869896137
Lasso Test RMSE: 3.0759219879077304
```

L2 Regularization:

```
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn.linear_model import Ridge
```

```
df_train = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')

df_train = df_train.dropna()
df_test = df_test.dropna()

x_train = df_train['x']
x_train = x_train.values.reshape(-1,1)
y_train = df_train['y']
y_train = y_train.values.reshape(-1,1)

x_test = df_test['x']
x_test = x_test.values.reshape(-1,1)
y_test = df_test['y']
y_test = y_test.values.reshape(-1,1)

ridge = Ridge()

ridge.fit(x_train, y_train)
print("Ridge Train RMSE:",
np.round(np.sqrt(metrics.mean_squared_error(y_train,
ridge.predict(x_train))), 5))
print("Ridge Test RMSE:",
np.round(np.sqrt(metrics.mean_squared_error(y_test,
ridge.predict(x_test))), 5))
```

Output:

```
Ridge Train RMSE: 2.80495
Ridge Test RMSE: 3.07131
```