School of Engineering and Applied Science (SEAS), Ahmedabad University

B.Tech (CSE Semester VI)/M.Tech/PhD:
Machine Learning (CSE 523)

Project Submission #2: Linear Regression

- **Group No.:** S_ECC4
- **Project Area:** Climate and Environment
- **Project Title:** Landslide Susceptibility Assessment by Novel Hybrid Machine Learning Algorithms
- **Name of the group members :**
  1. Jainam Chhatbar (1741002)
  2. Dhairya Dudhatra (1741058)
  3. Charmil Gandhi (1741059)
  4. Jinesh Patel (1741076)

1. **Implementation code:**

```
import pandas as pd
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
%matplotlib inline

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score

# Reading the dataset
df2 = pd.read_csv("ML_Regression_Data.csv", header='infer')

# Exracting input data and output data
x_axis = df2.iloc[:,1]
X = df2.iloc[:,2:]
Y = df2.iloc[:,0]

# Spliting the data into training and testing part
train_X, test_X, train_y, test_y, x_axis_train, x_axis_test =
    train_test_split(X, Y, x_axis, train_size=0.8, test_size=0.2)
```

```
# Merging the two matrix
data = np.column_stack((x_axis_test, test_X))
# Sorting the matrix by first column
data1 = data[data[:,0].argsort()]

# Extracting testing data from sorted matrix
x_axis_test = data1[:,0]
test_X = data1[:,1:]

### MLE using Normal equation, without Intercept

# For MLE (Maximum likelyhood estimation)
def max_lik_estimate(X, y):
  N, D = X.shape
  theta_ml = np.zeros((D, 1))          # Initiating the theta vector
  a = np.linalg.pinv(np.dot(X.T, X))  # pseudo inverse of (X^T)X
  b = np.dot(X.T, y)                   # Dot product of (X^T)y
  theta_ml = np.matmul(a, b)           # Multiplication of (X^T)X and (X^T)y
  return theta_ml

def predict_with_estimate(Xtest, theta):
  # predit the output. i.e, (X^T)theta
  prediction = np.dot(Xtest, theta)
  return prediction

# Estimating theta
theta = max_lik_estimate(train_X, train_y)

# Prediction of testing data
ml_prediction = predict_with_estimate(test_X, theta)

# ploting the results
plt.plot(x_axis, Y ,'+', markersize=10)
plt.plot(x_axis_test, ml_prediction)
plt.legend(["data", "maximum likelihood function"]);
plt.xlabel("Landslide susceptibility index (LSI) ")
plt.ylabel("No. of Lanslide")
plt.show()



### MLE using Normal equation, with intercept

def lr_mle(X, y):
  N, D = X.shape
  # augmented training inputs of size N x (D+1)
  aug_X = np.hstack([np.ones((N,1)), X])

  # new theta vector of size (D+1) x 1
```

```
    aug_theta = np.zeros((D+1, 1))

    augX_t = aug_X.transpose()      # Taking transpose of augmented X
    a = np.dot(augX_t,aug_X)        # Dot product
    b = np.linalg.pinv(a)           # pseudo inverse
    c = np.dot(b,augX_t)            # Dot product
    aug_theta_ml = np.dot(c,y)      # Calculating augmented theta
    return aug_theta_ml

def pred_mle(X, theta):
    y = np.dot(X, theta)            # Predicting output
    return y


N, D = test_X.shape
aug_X = np.hstack([np.ones((N,1)), test_X])

# Calculating theta
theta = lr_mle(train_X, train_y)

# Predicting theta
ml_pred = pred_mle(aug_X, theta)

# Ploting result
plt.figure()
plt.plot(x_axis_train, train_y, '+')
plt.plot(x_axis_test, ml_pred)
plt.xlabel("Landslide susceptibility index (LSI) ")
plt.ylabel("No. of Lanslide")
plt.legend(["data", "maximum likelihood function"]);
plt.show()



#### Polynomial regression

x = X
y = Y

# Converting X into nonlinear function
x_ = PolynomialFeatures(degree=3, include_bias=True).fit_transform(x)

# Applying regression on nonlinear function of X
model = LinearRegression(fit_intercept=False).fit(x_, y)

# Getting RMSE
r_sq = model.score(x_, y)
print('coefficient of determination:', r_sq)
```

```
# Predicting output from the trained model
y_pred = model.predict(x_)
print('predicted response:', y_pred, sep='\n')

# Plotting result
plt.plot(x_axis, y, '+')
plt.plot(x_axis, y_pred)
plt.xlabel("Landslide susceptibility index (LSI) ")
plt.ylabel("No. of Lanslide")
plt.legend(["data", "predicted function (degree of polynomial = 3)"]);
plt.show()


r2 = r2_score(y,y_pred)
print(r2)
```

2. **URL links:** Click here to go to our project drive

3. **Inference:**

Landslide is a natural phenomena, triggered by topographical, hydrological reasons. Landslides cannot be controlled or stopped, but the losses caused by it can be reduced by establishing a system which can predict landslides or area highly prone to landslides for better management. Usually 15 factors are used frequently to determine the susceptibility of landslides. Out of these 15 factors, we have just considered the 6 most significant factors to be used in our linear regression model to show relation between number of landslides and susceptibility index (LSI). Landslide susceptibility mapping is essential to describe the propensity of a landslide in a susceptible area. Such results can provide helpful information for the disaster managers, for urban planners, and decision makers in the landslide-prone area.

The classes used for susceptibility maps.

Susceptibility class	SI method		LR method	
	fifteen factors	six factors	fifteen factors	six factors
Extremely low	−12.31–-3.41	−3.83–−2.07	0.00–0.13	0.03–0.20
Low	−3.41–-2.17	−2.07–−1.20	0.13–0.32	0.20–0.38
Moderate	−2.17–-0.94	−1.20–−0.57	0.32–0.51	0.38–0.51
High	−0.94–0.23	−0.57–−0.04	0.51–0.67	0.51–0.64
Very high	0.23–1.33	−0.04–0.44	0.67–0.81	0.64–0.75
Extremly high	1.33–4.26	0.44–1.85	0.81–0.98	0.75–0.91

Figure 1: Susceptibility Class Table

15 factors considered :

- Elevation
- Slope angle
- Slope Aspect
- Total Curvature
- Profile Curvature
- Plan Curvature
- CTI (Hydrological Factor)
- SPI (Hydrological Factor)
- Drainage Density
- Distance to drainage network
- Lithology
- Density of geological boundaries
- Distance to geological boundaries
- Distance to faults
- NDVI (Normalized difference vegetation index)

The above mentioned 15 features have some sub-classes from which the subclass which had text inputs were mapped to integers accordingly.

We have selected the features using the Certainty Factor(CF) for that feature and value of CF varies between -1 and 1. Using this certainty factor(CF), Z value for every feature was found out.Once the CF values for classes of the causative factors are obtained, these factors are then incorporated pairwise using the combination rule:

$$Z = \begin{cases} CF1 + CF2\text{-}CF1CF2 & CF1, CF2 \geq 0 \\ CF1 + CF2 + CF1CF2 & CF1, CF2 < 0 \\ \frac{CF1+CF2}{1-\min(|CF1|, |CF2|)} & CF1,\ CF2,\ \text{opposite signs} \end{cases}$$

Figure 2: MLE With Intercept

The negative value of Z signifies: that particular feature does not contribute significantly in landslides' occuring and were considered for regression model. Out of these 15 factors, the 6 factors that were selected:

- Slope angle
- Slope Aspect
- Drainage Density
- Lithology
- Distance to geological boundaries
- Distance to faults

As multiple features are considered to determine the susceptibility of landslides, Multiple Linear Regression Model (LMR) was used. The value of each class corresponding to a feature (6 significant feature) were used to find the coefficient θ using Maximum Likelihood Estimation (MLE). The values of θ were further used in predictor function which would predict the number of landslides for a given Landslide Susceptibility Index (LSI). The regression model can be represented as :

$$\mathrm{P} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_6 x_6 + \epsilon$$

In above equation, θ_n represents coefficients of independent parameters and ϵ is error.

Firstly, we read the dataset from the given file and storing in the instance of panda. Then we split the data into training data and test data in 8:2. Since the data in test are chosen randomly from the original data, we sort the test data by LSI which the Land susceptibility Index. And then give the test data to prediction function.

4. **Results:**

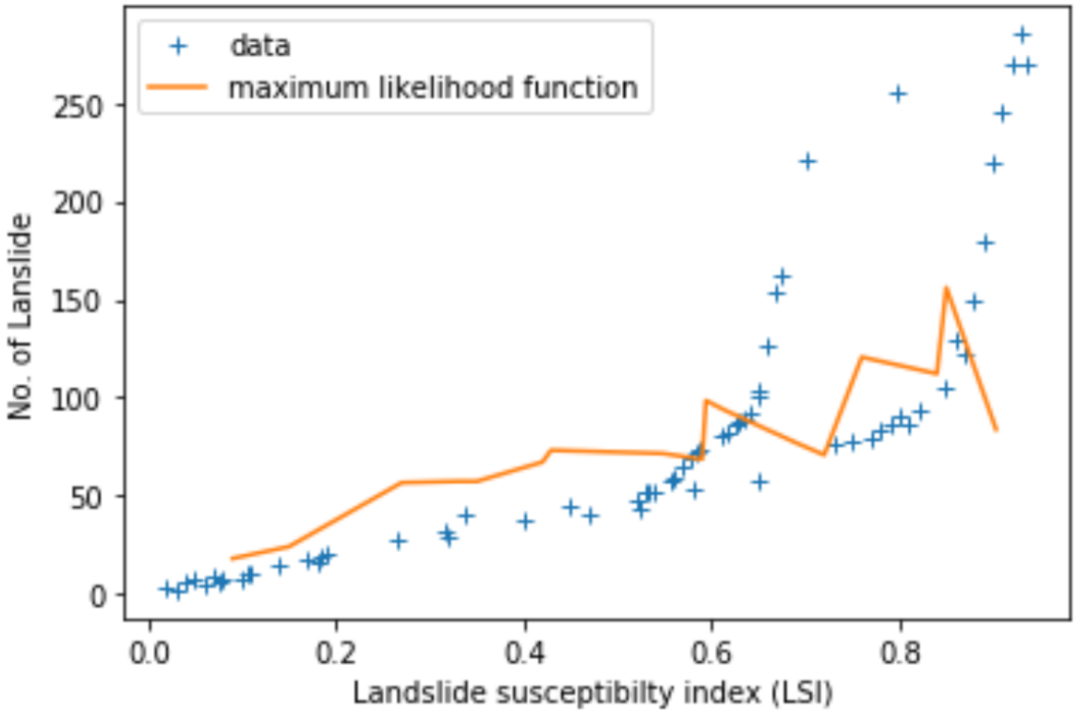


Figure 3: MLE With Intercept

The above figure represents the MLE calculation taking intercept into consideration. This signifies that the output y before we started the data will not always be 0. For this, it is always a good practice to consider the intercept.

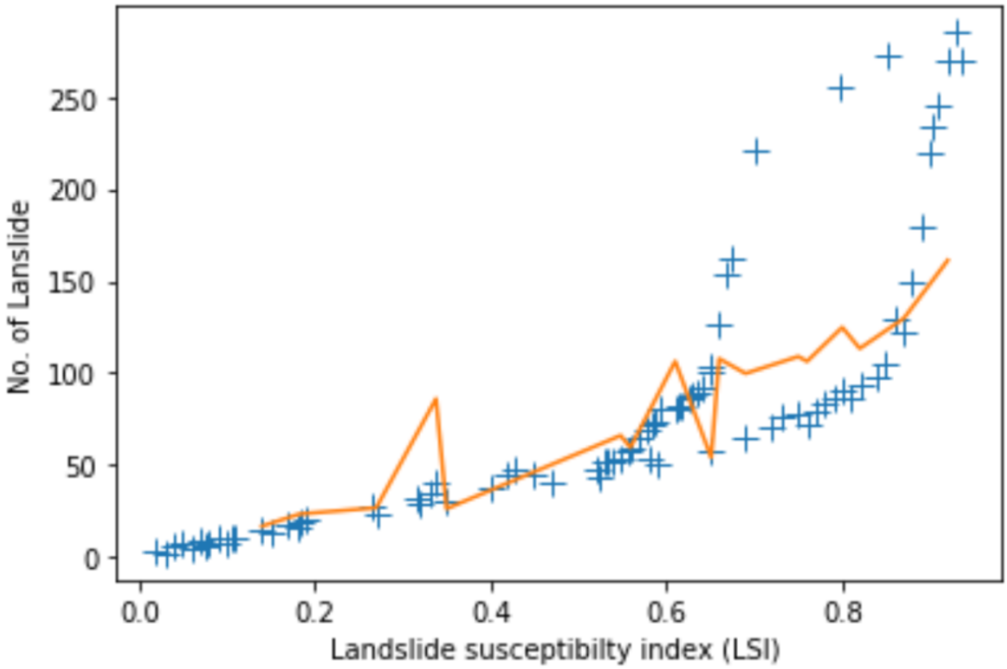
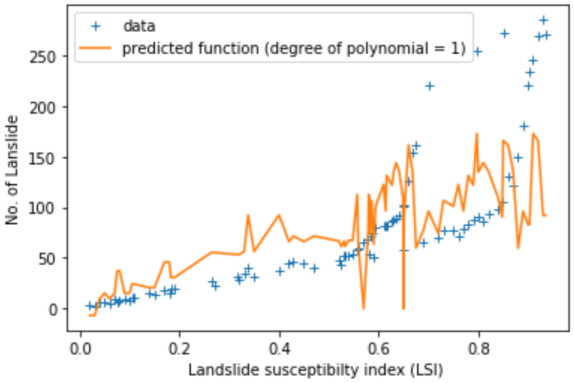


Figure 4: MLE Without Intercept

The above figure represents the MLE calculation without taking intercept into consideration. This signifies that the output y before we started the data will be 0. Thus the predicted data will always pass through origin.

2



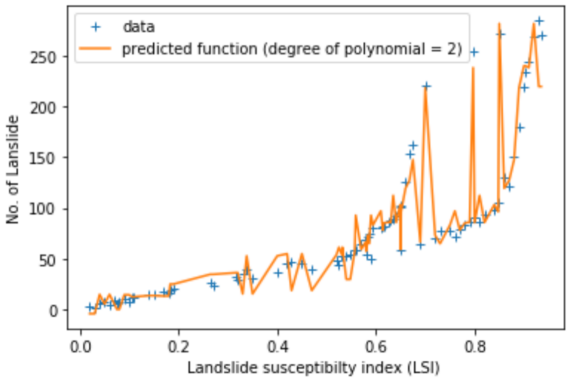
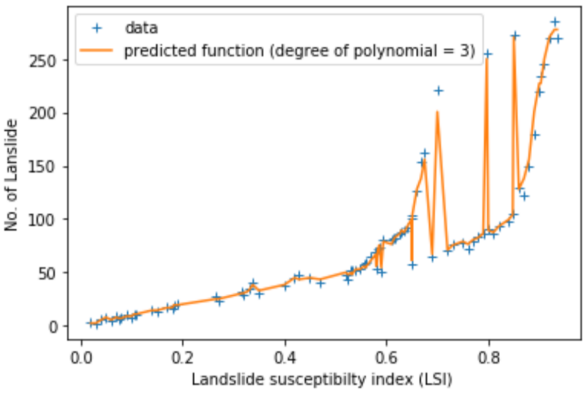


Figure 5: Polynomial Degree 1

Figure 6: Polynomial Degree 2

The above 2 figures are for polynomial with degree 1 and degree 2. As it can be seen from figure, for polynomial with degree 1, the predicted function results into much more haphazard data, where as polynomial with degree 2 performs better than polynomial with degree 2 but still fails to fit with original data. 2



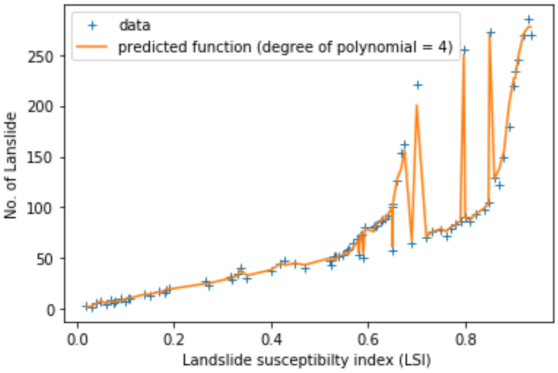


Figure 7: Polynomial Degree 3

Figure 8: Polynomial Degree 4

The above 2 figures are for polynomials with degree 3 and degree 4. As seen from figures above, the predicted function performs better with polynomial of degree 3 and 4 as the predicted values fit with the original data.

A prior knowledge of appropriate causative factors is required to successfully determine the landslide susceptibility. Positive value of Certainty Factor(CF) represents that those features have strong influence on landslide occurrence which increases the Landslide susceptibility Index of that place.