Slope Stability Prediction

Mini Project Report

Submitted by

Krishnapriya V M

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)
FEBRUARY 2022

DECLARATION

I, **Krishnapriya V M** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of our original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report titled **"Slope Stability Prediction"** submitted by **Krishnapriya V M** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2022.

Project Guide **Head of the Department**

Submitted for the viva-voice held on . . . . at . . . .

ACKNOWLEDGEMENT

We are deeply grateful to **Dr.Manoj George** , Principal, FISAT, Angamaly for providing us with adequate facilities, way and means by which we were able to complete our mini project work and I express my sincere thanks to **Dr. C. Sheela** ,Vice Principal FISAT, Angamaly.

Our sincere thanks to **Dr.Deepa Mary Mathwes**, Head of the department of MCA, FISAT, who had been a source of inspiration. I express heartiest thanks to **Mr.Sandhosh Kottam** our project guide for their encouragement and valuable suggestion . We express our heartiest gratitude to our scrum master **Dr.Shahana KU** and the faculty members in our department for their constant encouragement and never ending support throughout the project.We would also like to express our sincere gratitude to the lab faculty members for their guidance

Finally we express our thanks to all my friends who gave me wealth of suggestion for successful completion of this project.

ABSTRACT

Analysis and Prediction of slope stability are very important, because slope failure can lead to large disasters. With the development of economics, the number of slopes is increasing and landslides caused by slope instability have become one of the major disaster in the world. To reduce or prevent landslide damage, slope stability analysis and stabilization are required.

This project focused on the performance comparison of three supervised machine learning methods for slope stability prediction. Eight typical parameters such as regression coefficient, cohesion, factor of saftey, location, stability,slope intercept, slope height and pore water ratio- were chosen to predict the result. The Support vector machine(SVM), Naïve Bayes and Random Forest(RF) were proposed to establish classifiers. A dataset from domestic and abroad slope projects are established to train and test 3 classifiers. The result of accuracy is used to identify the best classifier and predict the stability of the slopes. K-fold cross validation is used for model comparison and parameter optimization.

Contents

Chapter 1

Introduction

In recent years, there are so many major disasters occurs in the world along due to the unstable slopes in so many countries. This affect on the people's lives and also they are losing their livelihood. It is imperative to conduct a study on the stability of the slopes and also if necessary ,it is important to find appropriate solutions.

In this project, we focused on the slope stability of domestic slope projects using supervised learning methodologies and based on the dependent parameters of the slope we are predicting the stability and then if any chance to occur any disasters we can take necessary actions according to the predicted output.

Chapter 2

PROOF OF CONCEPT

The analysis and prediction of slope stability are very important, because slope failure can lead to large disasters. By definition, slope stability is a measure of how resistant a natural or man-made slope is to failure due to collapse or sliding. It is an important consideration in the management of Economy. With the development of economics, the number of slopes is increasing and landslides caused by slope instability have become one of the major disasters in the world. To reduce to prevent landslide damage, slope stability analysis and stabilization are required.

This project focused on the performance comparison of three supervised machine learning methods for slope stability prediction. Eight typical parameters such as regression coefficient, cohesion, factor of saftey, location, stability,slope intercept, slope height and pore water ratio- were chosen to predict the result. The support vector machine(svm), naive bayes and random forest(rf) were proposed to establish classifiers. A dataset from domestic and abroad slope projects are established to train and test 3 classifiers. The result of accuracy is used to identify the best classifier and predict the stability of the slopes. Kfold cross validation is used for model comparison and parameter optimization

Chapter 3

IMPLEMENTATION

In this project, implement performance comparison of three supervised machine learning methods for slope stability prediction. Eight typical parameters such as regression coefficient, cohesion, factor of saftey, location, stability,slope intercept, slope height and pore water ratiowere chosen to predict the result. The Support vector machine(SVM), Naïve Bayes and Random Forest(RF) were proposed to establish classifiers. A dataset from domestic and abroad slope projects are established to train and test 3 classifiers. The result of accuracy is used to identify the best classifier and predict the stability of the slopes. K-fold cross validation is used for model comparison and parameter optimization.

3.1 System Architecture

3.2 Dataset

Slope project datasets from different domestic and abroad areas were used to train the model. Split the dataset into train set and test set to predict the model with maximum accuaracy

3.3 Modules

3.3.1 DATA PREPROCESSING

Data preprocessing is an iterative process for the transformation of the raw data into understandable and useable forms. The preprocessing is essential to handle the missing values and inconsistencies.

3.3.2 FEATURE EXTRACTION

When performing analysis of complex data one of the major problems occurs from the number of variables involved. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

3.3.3 TRAINING THE MODEL

The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model

3.3.4 EVALUATION

Model evaluation techniques in machine learning are helping us to find a better model among all other models in machine learning.

3.4 ALGORITHM

RANDOM FOREST

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, ”Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.” Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

How it Works?

Step-1: select random K data points from the training set.
Step-2: build the decision trees associated with the selected data points (subsets).
Step-3: choose the number n for decision trees that you want to build.
Step-4: repeat step 1 2
Step Step-5: for new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

NAIVE BAYES

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.
It is based on the Baye's theorem: Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

Chapter 4

RESULT ANALYSIS

The result of the proposed project Slope Stability Prediction lies in developing a handy web app that can successfully predict whether the slopeis stable or not according to the data given by the user.

The proposed system takes the value such as height of the slope, Angle of friction, Cohesion and pore-water ratio user information and generate predictions on it

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The analysis and prediction of slope stability are very important, because slope failure can lead to large disasters. By definition, slope stability is a measure of how resistant a natural or man-made slope is to failure due to collapse or sliding. It is an important consideration in the management of Economy. Here we presented a model to detect the stability of the slope according to it's attribute's values. The classification accuracy of the proposed model is 82.%. which is the highest achieved accuracy to the best of our knowledge on the data sets used in the experiments.

The performance of Random Forest is shown to be better than others in terms of accuracy.Our future goal is to overcome the disasters due to instability of the slope. The proposed model helps to identify the defects and take necessary actions in future.

5.2 Future Scope

By analysing the stability of the slope:

- Avoid disasters due to instability and save the people's lives.
- Can handle larger deformations.

.

Chapter 6

CODING

6.1 Training.ipynb

1. **Mount Drive**

```
from google.colab import drive
drive.mount('/content/drive')
```

2. **Importing necessary libraries**

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn import metrics,svm
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
```

from sklearn.linear$_{m}odelimportLinearRegression, LogisticRegression$
$importmatplotlib.pyplotasplt$
$fromsklearn.metricsimportconfusion_{m}atrix, classification_{r}eport$
$fromsklearn.preprocessingimportStandardScaler, PolynomialFeatures$
$fromsklearn.metricsimportroc_{c}urve$
$fromsklearn.metricsimportroc_{a}uc_{s}core$
$frommatplotlibimportpyplot$
$importseabornassn$

print('Library Importing Complete')

3. **Loading data**

data =pd.read$_{c}sv("D:3Project/slope.csv")$

Split data into Training and Testing set

$fromsklearn.model_{s}electionimporttrain_{t}est_{s}plit$
$X_{t}rain, X_{t}est, y_{t}rain, y_{t}est = train_{t}est_{s}plit(X, Y, test_{s}ize = 0.25, random_{s}tate = 0)$

Data Preprocessing

X$_{t}emp = X$
$scaler = StandardScaler()$
$X = scaler.fit_{t}ransform(X)$
$sns.violinplot(y = data["H"])$
$sns.violinplot(y = data[""])$
$sns.violinplot(y = data[""])$
$sns.violinplot(y = data["C"])$

$sns.violinplot(y = data[""])$
$ax3 = sns.violinplot(y = data["ru"])$
$corrMatrix = X.corr()$
$heat_map = sn.heatmap(corrMatrix, annot = True)$

5. **Creating Evaluating Models**
from sklearn.svm import SVC
cost= 1, 2, 4, 8, 16, 32
sigma= 0.0025, 0.005, 0.01, 0.015, 0.02, 0.025, 0.25, 1
for c in cost:
svc=SVC(C = c)
svc.fit(X$_train, y_train$)
$pred = svc.predict(X_test)$

score = metrics.accuracy$_score(y_test, pred)$
$print(score)$
$print(c)$

svc = SVC(kernel = 'linear')
svc.fit(X$_train, y_train$)
$pred = svc.predict(X_test)$

pred=svc.predict(X$_test$)
$score = metrics.accuracy_score(y_test, pred)$
$print("accuracy :$
$metrics.plot_confusion_matrix(svc, X_test, y_test)$

ns$_probs = [0 for_in range(len(y_test))]$

$model = SVC(kernel =' linear', probability = True)$
$model.fit(X_t rain, y_t rain)$
$score = metrics.accuracy_s core(y_t est, pred)$
$print("accuracy :$
$metrics.plot_c onfusion_m atrix(svc, X_t est, y_t est)$
$lr_p robs = model.predict_p roba(X_t est)$
$lr_p robs = lr_p robs[:, 1]$
$ns_a uc = roc_a uc_s core(y_t est, ns_p robs)$
$lr_a uc = roc_a uc_s core(y_t est, lr_p robs)$
$print('SVC : AUC =$
$ns_f pr, ns_t pr,_= roc_c urve(y_t est, ns_p robs)$
$lr_f pr, lr_t pr,_= roc_c urve(y_t est, lr_p robs)$
$pyplot.plot(ns_f pr, ns_t pr, linestyle =' --', label =' NoSkill')$
$pyplot.plot(lr_f pr, lr_t pr, marker =' .', label =' SVC')$
$pyplot.xlabel('FalsePositiveRate')$
$pyplot.ylabel('TruePositiveRate')$
$pyplot.legend()$
$pyplot.show()$

from sklearn.svm import SVC classifier = SVC(kernel = ’linear’, random$_s tate = 0)$
$classifier.fit(X_t rain, y_t rain)$
$y_p red = classifier.predict(X_t est)$
$y_p red$
$fromsklearn.metricsimportconfusion_m atrix$
$fromsklearn.metricsimport$
$classification_r eport$
$print(classification_r eport(y_t est, y_p red))$
$print(confusion_m atrix(y_t est, y_p red))$

$fromsklearn.ensembleimportRandomForestClassifier$

$rf = RandomForestClassifier(max_{d}epth = 2, random_{s}tate = 0)$
$rf.fit(X_{t}rain, y_{t}rain)$
$metrics.plot_{r}oc_{c}urve(rf, X_{t}est, y_{t}est)$

ns$_{p}robs = [0for_{i}nrange(len(y_{t}est))]$
$model = RandomForestClassifier(max_{d}epth = 2, random_{s}tate = 0)$
$model.fit(X_{t}rain, y_{t}rain)$
$lr_{p}robs = model.predict_{p}roba(X_{t}est)$
$lr_{p}robs = lr_{p}robs[:, 1]$
$ns_{a}uc = roc_{a}uc_{s}core(y_{t}est, ns_{p}robs)$
$lr_{a}uc = roc_{a}uc_{s}core(y_{t}est, lr_{p}robs)$
$print('RandomForest : AUC =$
$ns_{f}pr, ns_{t}pr,_{=} roc_{c}urve(y_{t}est, ns_{p}robs)$
$lr_{f}pr, lr_{t}pr,_{=} roc_{c}urve(y_{t}est, lr_{p}robs)$
$pyplot.plot(ns_{f}pr, ns_{t}pr, linestyle =' --', label =' NoSkill')$
$pyplot.plot(lr_{f}pr, lr_{t}pr, marker =' .', label =' RandomForest')$
$pyplot.xlabel('FalsePositiveRate')$
$pyplot.ylabel('TruePositiveRate')$
$pyplot.legend()$
$pyplot.show()$

from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(n$_{e}stimators = 10, criterion = "entropy")$
$classifier.fit(X_{t}rain, y_{t}rain)$
$y_{p}red = classifier.predict(X_{t}est)$
$fromsklearn.metricsimportconfusion_{m}atrix$
$fromsklearn.metricsimportclassification_{r}eport$
$print(classification_{r}eport(y_{t}est, y_{p}red))$
$print(confusion_{m}atrix(y_{t}est, y_{p}red))$

X=data.iloc[:,0:-2]
y=data.iloc[:,-2]
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X$_scaled = scaler.fit_transform(X)$
$fromsklearn.model_selectionimporttrain_test_split$

X$_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size = 0.5, random_state =$
$1)$
$fromsklearn.naive_bayesimportGaussianNBclassifier = GaussianNB()$
$classifier.fit(X_train, y_train)$

y$_pred = classifier.predict(X_test)$
y_pred

from sklearn.metrics import confusion$_matrix$
$fromsklearn.metricsimportclassification_report$
$print(classification_report(y_test, y_pred))$
$print(confusion_matrix(y_test, y_pred))$

6.2 app.py

import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render$_template$
$importpickle$

app = Flask($_{name_{,static_url path=\text{""},static_folder=\text{"static"})model=pickle.load(open('model.pkl','rb'))}}$

@app.route('/')
def home():
return render$_{t}emplate('index.html')$

@app.route('/predict',methods=['POST'])

def predict():

height = float(request.form['height'])
Cohesion = float(request.form['Cohesion'])
angleoffriction = float(request.form['angleoffriction'])
porewaterratio = float(request.form['porewaterratio'])

prediction = model.predict([[height,Cohesion,angleoffriction,porewaterratio]])

if prediction[0] == 1:
pred = "Slope is not stable, take necessary actions"
else:
pred = "Slope is Stable."

return render$_{t}emplate('index.html',$
$prediction_{t}ext = pred)$

if ${}_{name_{==''}{}_{main_{'':app.run(debug=True)}}}$

6.3 model.py

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

import pickle
data= pd.read$_c sv('slope.csv')$

dataset$_X = data.iloc[:,[1,3,4,5]].values$
$dataset_Y = data.iloc[:,6].values$

print(dataset$_X$)

X = dataset$_X$
$Y = dataset_Y$

from sklearn.model$_s election import train_t est_s plit$
$X_t rain, X_t est, Y_t rain, Y_t est = train_t est_s plit(X, Y, test_s ize = 0.20, random_s tate = 42, stratify = data['Stability'])$

from sklearn.ensemble import RandomForestClassifier
rf$_c lf = RandomForestClassifier(max_f eatures ==' auto')$
$rf_c lf.fit(X_t rain, Y_t rain)$

ypred=rf$_{c}lf.predict(X_{t}est)$

rf$_{c}lf.score(X_{t}est, Y_{t}est)$

new$_{d}ata = [[35.0, 15.0, 0.0, 31.0]]$
$pred = rf_{c}lf.predict(new_{d}ata)$
$pred[0]$

pickle.dump(rf$_{c}lf, open('model.pkl',' wb'))$
$model = pickle.load(open('model.pkl',' rb'))$

Chapter 7

SCREEN SHOTS

Here I add some sample screenshots of the proposed system which includes,

- Home Screen
- Prediction

Figure 7.1: Home Screen

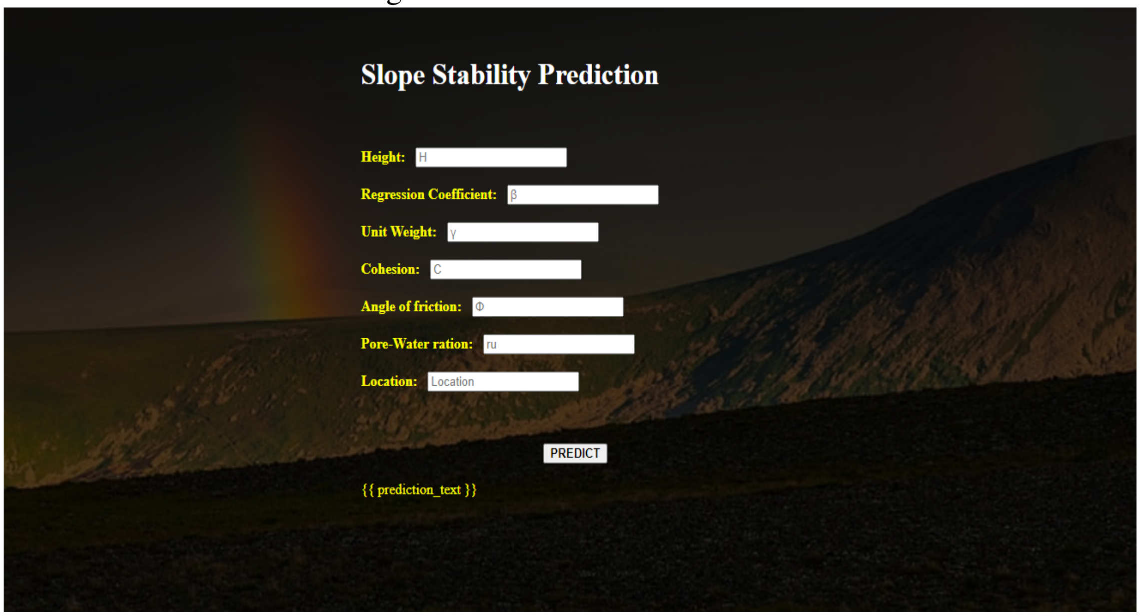
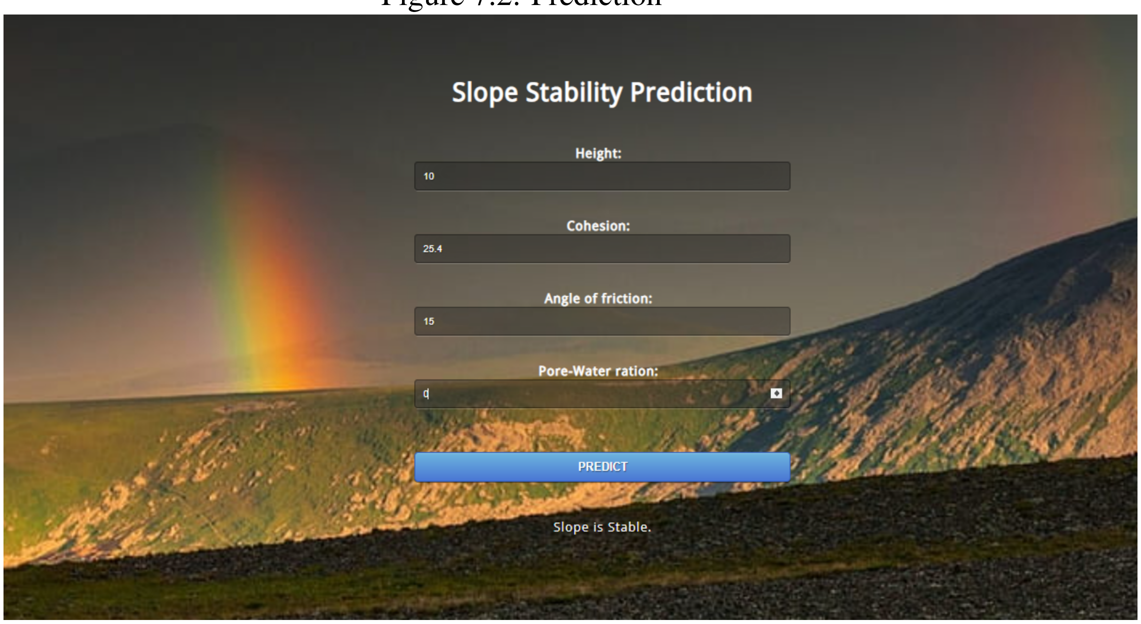


Figure 7.2: Prediction



Chapter 8

REFERENCES

(a) www.google.com

(b) www.youtube.com

(c) www.wikipedia.com

(d) https://link.springer.com/article/10.1007/s10706-004-8680-5

(e) https://ieeexplore.ieee.org/abstract/document/8383970