PROJECT SCHOOL CERTIFICATE

1. **Name:** MAIDAMPALLIY AJAY KUMAR
2. **ID No:** 21BD1A050U
3. **Section:** CSE-A
4. **Title:** Flood Detection
5. **Technology used:**

   Machine Learning:
   - Logistic Regression
   - Support Vector Machines(SVM)
   - Random Forest
   - XGBoost

   Flutter

6. **Faculty Incharge:** Dr.S.Rajashekhar

Student Signature: **Faculty Signature:**

Date of Issue: 19/11/2022 **Date of submission:8/2/2023**

PROJECT DESCRIPTION

Urban flooding is significantly different from rural flooding as urbanization leads to developed catchments, which increases the flood peaks from 1.8 to 8 times and flood volumes by up to 6 times.

Consequently, flooding Occurs very quickly due to faster flow times (in a matter of minutes).

Urban areas are densely populated and people living in vulnerable areas suffer due to flooding, sometimes resulting in loss of life.

It is not only the event of flooding but the secondary effect of exposure to infection also has its toll in terms of human suffering, loss of livelihood and, in extreme cases, loss of life. Increasing trend of urban flooding is a universal phenomenon and poses a great challenge to urban planners the world over.

Predicting the extent of calamity requires lots of parameters like rainfall, vegetation etc.

Improper disposal of solid waste, including domestic, commercial and industrial waste and dumping of construction debris into the drains also contributes significantly to reducing their capacities.

It is imperative to take better operations and maintenance actions. The main objective of this problem is to help the city managers or urban residents in for predicting the occurrence of urban flooding and can help them to take preventive

DATASET DESCRIPTION

➔ We have used Kerala Dataset for this project

➔ It consists of the data about annual and monthly rainfall in Kerala between the years 1900-2018 and occurrence of flood in those years.

SUBDIVISI	YEAR	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	ANNUAL I	floods
KERALA	1901	28.7	44.7	51.6	160	174.7	824.6	743	357.5	197.7	266.9	350.8	48.4	3248.6	YES
KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205	315.8	491.6	358.4	158.3	121.5	3326.6	YES
KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157	59	3271.2	YES
KERALA	1904	23.7	3	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES
KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO
KERALA	1906	26.7	7.4	9.9	59.4	160.8	414.9	954.2	442.8	131.2	251.7	163.1	86	2708	NO
KERALA	1907	18.8	4.8	55.7	170.8	101.4	770.9	760.4	981.5	225	309.7	219.1	52.8	3671.1	YES
KERALA	1908	8	20.8	38.2	102.9	142.6	592.6	902.2	352.9	175.9	253.3	47.9	11	2648.3	NO
KERALA	1909	54.1	11.8	61.3	93.8	473.2	704.7	782.3	258	195.4	212.1	171.1	32.3	3050.2	YES
KERALA	1910	2.7	25.7	23.3	124.5	148.8	680	484.1	473.8	248.6	356.6	280.4	0.1	2848.6	NO
KERALA	1911	3	4.3	18.2	51	180.6	990	705.3	178.6	60.2	302.3	145.7	87.6	2726.7	NO
KERALA	1912	1.9	15	11.2	122.7	217.3	948.2	833.6	534.4	136.8	469.5	138.7	22	3451.3	YES
KERALA	1913	3.1	5.2	20.7	75.7	198.8	541.7	763.2	247.2	176.9	422.5	109.9	45.8	2610.8	NO
KERALA	1914	0.7	6.8	18.1	32.7	164.2	565.3	857.7	402.2	241	374.4	100.9	135.2	2899.1	NO
KERALA	1915	16.9	23.5	42.7	106	154.5	696.1	775.6	298.8	396.6	196.6	302.5	14.9	3024.5	YES
KERALA	1916	0	7.8	22	82.4	199	920.2	513.9	396.9	339.3	320.7	134.3	8.9	2945.3	YES
KERALA	1917	2.9	47.6	79.4	38.1	122.9	703.7	342.7	335.1	470.3	264.1	256.4	41.6	2704.8	NO

MODEL DESCRIPTION

1.Logistic Regression:

A logistic regression model predicts a dependant data variable by analysing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives.

2.Random Forest:

Random Forest model grows and combines multiple decision trees to create a “forest.” A decision tree is another type of algorithm used to classify data. In very simple terms, you can think of it like a flowchart that draws a clear pathway to a decision or outcome; it starts at a single point and then branches off into two or more directions, with each branch of the decision tree offering different possible outcomes.

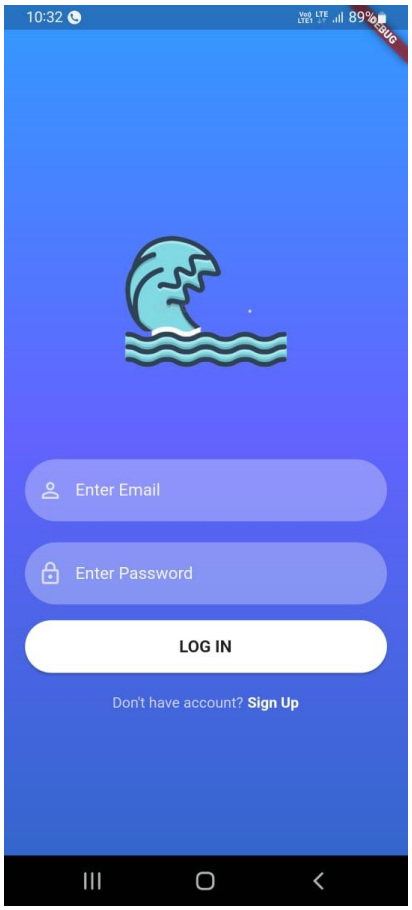
3.Support Vector Machine;

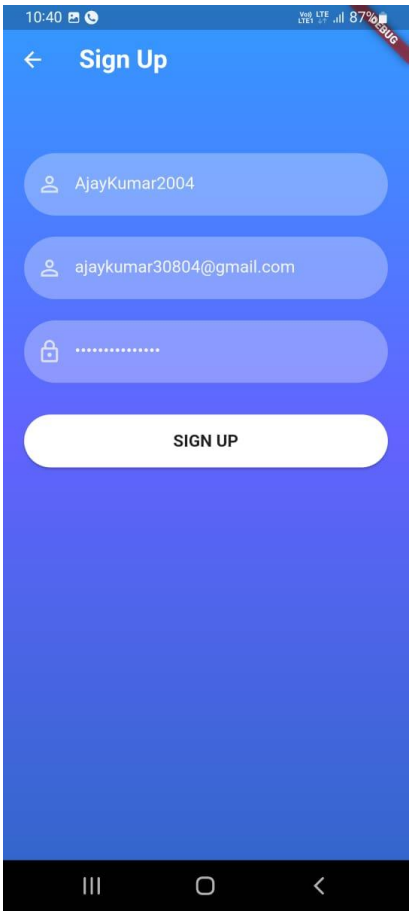
SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

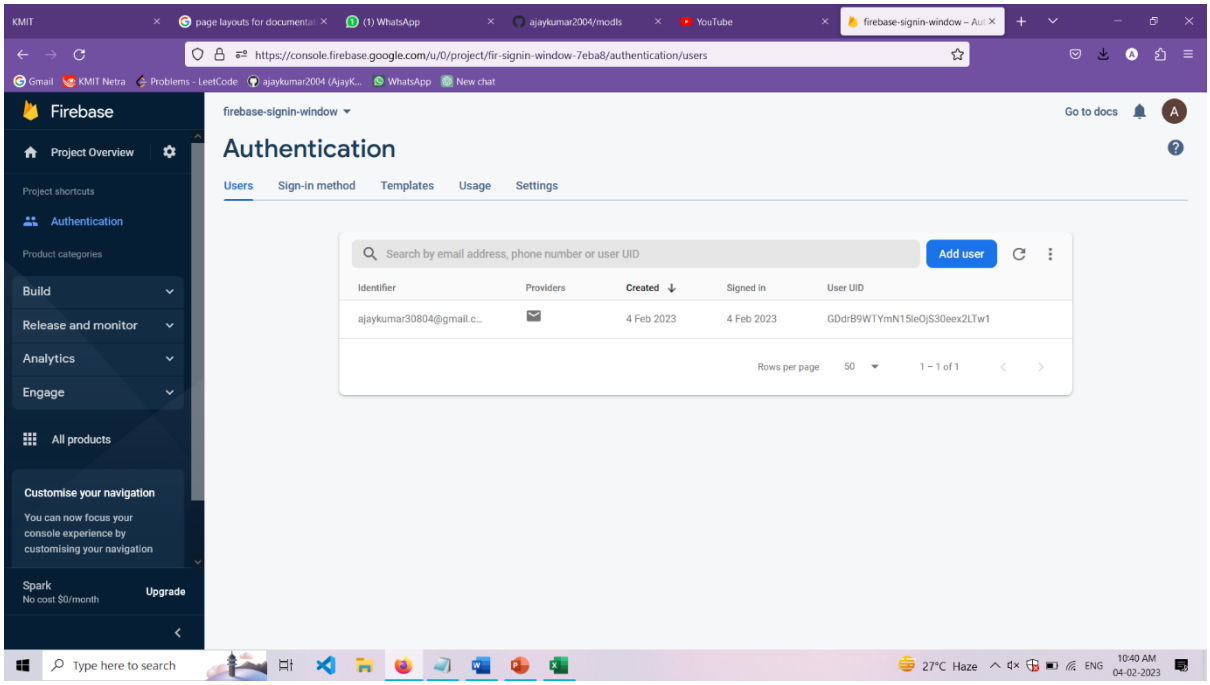
4.XG Boost:

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

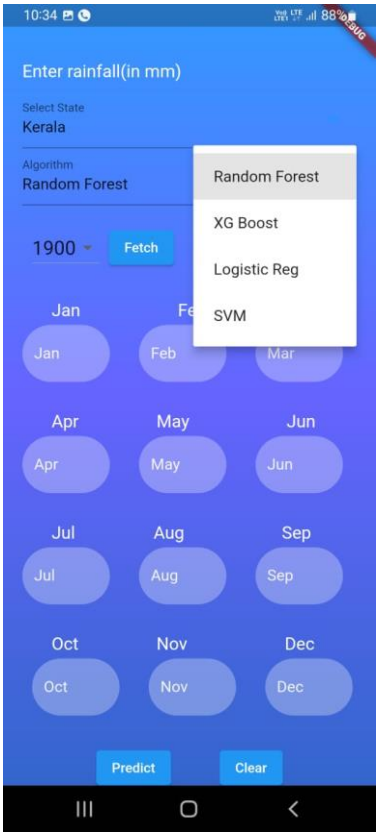
SCREENSHOTS

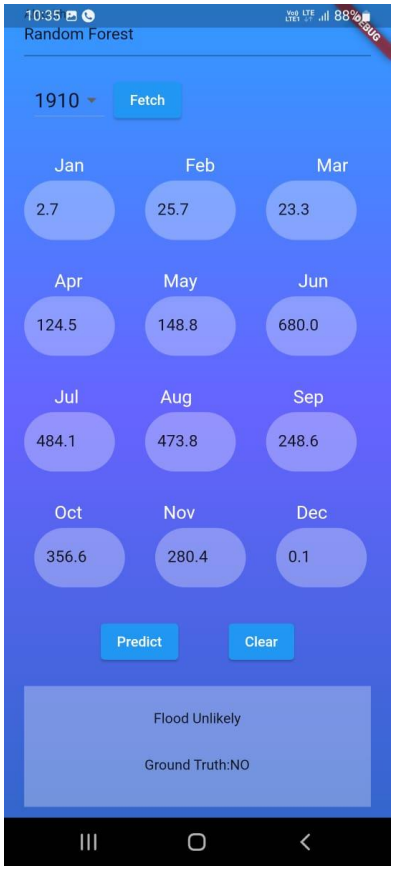


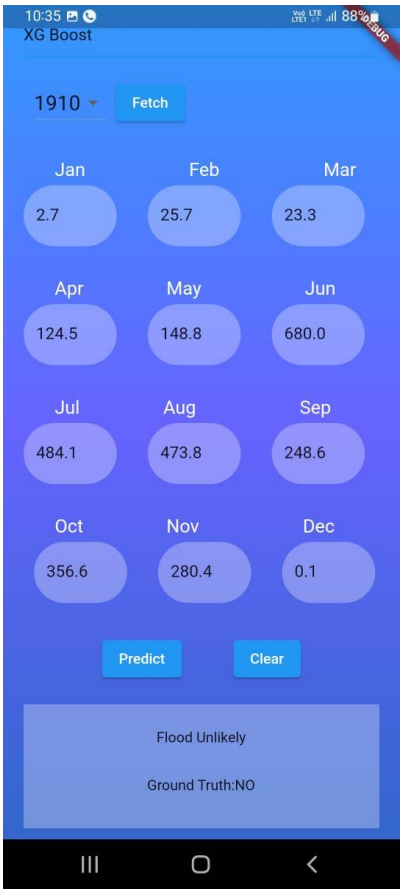


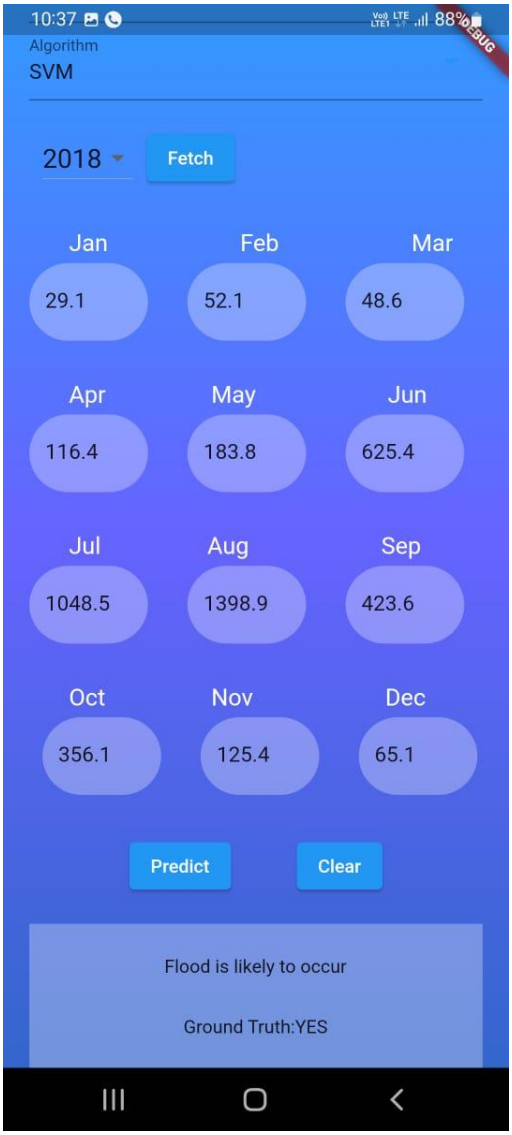



10:34
88%
(in mm)
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
Fetch
Feb
Mar
May
Jun
Aug
Sep
Nov
Dec
Predict
Clear


10:34
88%
Enter rainfall(in mm)
Select State
Kerala
Algorithm
Random Forest
Random Forest
XG Boost
Logistic Reg
SVM
1900
Fetch
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Predict
Clear


10:35
88%
Random Forest
1910
Fetch
Jan
Feb
Mar
2.7
25.7
23.3
Apr
May
Jun
124.5
148.8
680.0
Jul
Aug
Sep
484.1
473.8
248.6
Oct
Nov
Dec
356.6
280.4
0.1
Predict
Clear
Flood Unlikely
Ground Truth:NO


10:35
88%
XG Boost
1910
Fetch
Jan
Feb
Mar
2.7
25.7
23.3
Apr
May
Jun
124.5
148.8
680.0
Jul
Aug
Sep
484.1
473.8
248.6
Oct
Nov
Dec
356.6
280.4
0.1
Predict
Clear
Flood Unlikely
Ground Truth:NO



Model Accuracy:

```
print("XGB accuarcy :",loaded2.score(x_test,y_test))
print("Random Forest accuarcy :",loaded3.score(x_test,y_test))
print("Logistic Rgression accuracy :",loaded4.score(x_test,y_test))
print("SVM accuracy :",loaded1.score(x_test,y_test))
```

✓ 0.1s

```
XGB accuarcy : 0.8333333333333334
Random Forest accuarcy : 0.875
Logistic Rgression accuracy : 0.7916666666666666
SVM accuracy : 0.875
```